Synchronous Measurements and Power Quality Monitoring

Volker Kuhlmann

A thesis presented for the degree of

Doctor of Philosophy
in Electrical and Electronic Engineering

21 April 2004

University of Canterbury
Christchurch
New Zealand

Supervisors:
Dr C. P. Arnold
Assoc Prof M. B. Dewe

Synchronous Measurements and Power Quality Monitoring

Volker Kuhlmann

Email: VolkerKuhlmann@gmx.de

All photos: Volker Kuhlmann

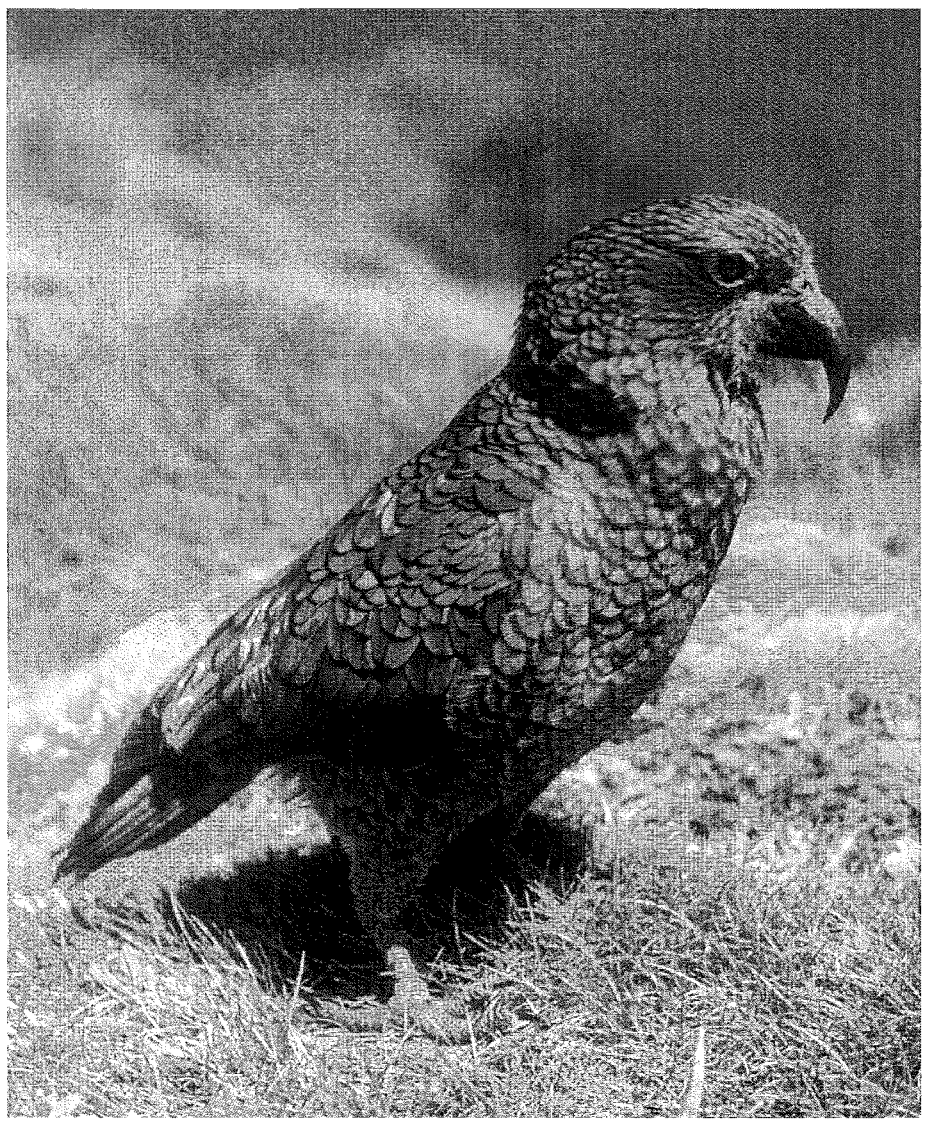


Photo 1: Kea, *Nestor notabilis*. See page 2.

Abstract

The task of simultaneous power systems measurements at locations which can be hundreds of kilometres apart carries the problem of precise synchronisation. To introduce this topic, this thesis begins with an overview of time, time stamping and synchronised data acquisition, and reviews its application to power quality monitoring. It then discusses the requirements for such systems and how different application scenarios shift the emphasis between aspects of the requirements. The complexity of a power distribution monitoring system is most dependent on the number of channels and sites which must be analysed, and the required time stamping accuracy. For some applications, samples need to be time stamped with an accuracy of 1 µs. A requirements specification template is presented which aids, for example, in purchase decisions to establish needed features.

A specific example of such a system, CHART III, has been developed at the University of Canterbury (Christchurch, New Zealand), which uses a hardware time base and sample clock generation, hardware time stamping, and GPS synchronisation to achieve a time stamping accuracy of 0.5 µs. The design of the time base of this system is published in this thesis and described in detail.

The CHART III system was used to gain practical experience and to establish its usability and operational limitations, and provided input for the theoretical considerations of an ideal system. Synchronised distributed data acquisition using two and three CHART III instrument was performed on two live power systems, collecting data in the frequency and time domains. A number of enhancements were made as a result, particularly to the control and analysis software in the areas of extending the handling of the GPS receiver and provision of additional system status and error information. Because the emphasis of the work is on the instrumentaton, no further analysis of the collected data is presented in this thesis.

The CHART III system was connected to the internet to investigate issues of remote configuration and the consolidation of sample analysis at a single powerful computer. Limits for time domain measurements, which have a higher data rate than the system can handle continuously, were established as being a minimum of 10 seconds.

In a data acquisition system, the quantisation error introduced by the ADC sets a lower limit for the noise. The effects of this quantisation noise on the recovery of harmonic magnitudes and phases were examined. Simulations were performed to model the influence of ADC width, fast Fourier transform length, harmonic amplitude and harmonic order. Both magnitude and phase errors are independent of harmonic order, decrease with the number of ADC quantisation levels, and decrease with the square root of the transform length. The magnitude error is independent of the harmonic amplitude for a sufficiently large amplitude to noise ratio. The phase error is inversely proportional to the amplitude. The accuracy of a harmonic analyser can therefore be increased by increasing the ADC width or the transform length. For accuracies likely to be required by typical power quality applications, these simulations indicate that a 12 bit ADC gives sufficiently accurate results.

Finally, the effect of current trends in microprocessor technology is discussed. Power quality monitoring systems can now be built much more simply and cheaply then when CHART III was designed. The most important improvement is that a single standard CPU can now handle the data from a number of channels, eliminating the need for specialised digital signal processors and the associated cost of producing software for a second architecture. Processor performance seems to be set to increase steadily, promising future improvements in time stamping accuracy, the number of channels which can be handled by one processor, and the availability of more complex analysis functions.

Keywords

time transfer
time base
time stamping
time tagging
ADC quantisation error
data acquisition
data processing
distributed data acquisition
synchronised data acquisition
microprocessors
power quality
synchronisation
power system harmonics
power system measurements
power system monitoring
analog-digital conversion
discrete Fourier transforms
local area networks

Acknowledgements

The CHART III project was a group endeavour, and I would like to thank the other members of the team, in particular Mike Dewe as project manager, and Jos Arrillaga for his skilled leadership of power systems research at the university. I owe special thanks to my supervisors, Mike Dewe and later in the project Chris Arnold, for their help, encouragement and patience.

The support of Transpower New Zealand Ltd in the early stages of this research is gratefully acknowledged. The field tests described in this thesis were undertaken with the help of Southpower Ltd and Transpower New Zealand Ltd.

Alastair Sinton and Michael Cree made their computers available for running fftw benchmarks, and Michael proofread the resulting text (section 11.2). Alastair's enthusiam for digital signal processing theory, and his knowledgable assistance with the error simulations described in chapter 10, greatly helped with that area of the work.

Finally, I would like to thank my friends and flatmates for putting up with the long periods when I did little but sit in front of the computer, and particularly Rebecca, whose encouragement, support and proofreading skills were invaluable.

Contents

Abstract vii

Acknowledgements ix

Glossary xxiii

1 Introduction 1
- 1.1 Introduction 1
- 1.2 Data Acquisition Systems Overview 1
- 1.3 The Requirement for Precise Synchronisation 2
- 1.4 Literature Overview 3
- 1.5 Thesis Overview 3

2 Time and Synchronised Data Acquisition 5
- 2.1 Time Keeping and Time Scales 5
- 2.2 Time Transfer 9
  - 2.2.1 Satellite-Based Navigation Systems 12
- 2.3 Synchronised Data Acquisition 13

3 Review of Power Quality Monitoring 17
- 3.1 Introduction 17
- 3.2 Phasor Measurements 17
- 3.3 Power Quality 18
- 3.4 Locating Faults on Transmission Lines 19
- 3.5 Accuracy Requirements 20

4 System Requirements for Power Quality Instrumentation 23
- 4.1 System Aspects 24
- 4.2 Analog-to-Digital Converters 26
- 4.3 Input Stage 28
- 4.4 Sampling and Sample Clock Generation 29
- 4.5 Time Stamping 30
- 4.6 Software 31
- 4.7 Designing Custom Hardware 33

5 System Requirements Specification Template 35
- 5.1 System Overview 35
- 5.2 System Configuration Considerations 36
  - 5.2.1 System Complexity 36
  - 5.2.2 Duration of Measurements 37
  - 5.2.3 Electromagnetic Environment Considerations 37
  - 5.2.4 Communications 38
  - 5.2.5 Separation of the Main Data Processing Unit from ADC Stage 38
  - 5.2.6 Hardware vs. Software Implementations 38

5.2.7 Power Supply 39
5.2.8 Development Constraints 39
5.2.9 Compliance With Relevant Standards 40
5.2.10 System Calibration 40
5.2.11 Overall Volume and Weight 40
5.2.12 Reliability 40
5.2.13 Hardware Resources 41
5.3 Data Acquisition Modules 41
5.3.1 Input Stage 41
5.3.2 Analog-to-Digital Converters 42
5.3.3 Sampling 42
5.3.4 Time Stamping 43
5.3.5 General-Purpose Digital Inputs and Outputs 44
5.3.6 Shielding / EMC 44
5.4 Data Processing and Software 44
5.4.1 CPU 45
5.4.2 Built-in Functions 45
5.4.3 User-Definable Functions 45
5.4.4 Trigger Conditions 45
5.4.5 Data Storage 46
5.4.6 Compensation for Sensor Characteristics 46
5.5 Communications 46
5.5.1 Remote Management 46
5.5.2 Interaction With Other Monitoring and Control Equipment 47
5.5.3 Control by External Signal 47
5.5.4 Encryption 47
5.6 User Interface 47
5.6.1 Instrument Controls 47
5.6.2 Instrument Setup 48
5.6.3 Display 48
5.6.4 Documentation 48
5.7 Environmental Considerations 49
5.7.1 Physical Environment 49
5.7.2 Electromagnetic Environment 49
5.7.3 Galvanic Isolation 49
5.7.4 Overvoltage Protection 49
5.8 Maintenance and Design Considerations 50
5.8.1 Software 50
5.8.2 Hardware 50

6 CHART III System Design 51
6.1 System Components 51
6.1.1 The Parallel Processing Unit (PPU) 51
6.1.2 The Remote Data Capturing Module (RDCM) 53
6.1.3 The Data Acquisition and Processing Module (DAPM) 53
6.1.4 The Digital Services Module (DSM) 54
6.1.5 The Control and Display Unit (CADU) 54
6.2 System Integration Aspects 54
6.2.1 Coherent Sampling and the SRM 55
6.2.2 Distributing a Precision Time Throughout the System 56
6.3 Consideration of Timing Errors 56
6.4 Conclusion 57

7 **The Design of the Time Base** 59
7.1 Overview 59
7.2 Modular Interface eXtension (MIX) 60
7.3 The TMS320C31 Digital Signal Processor 61
7.4 The GPS Receiver Interface 63
7.5 The Fibre-Optic Interface 63
7.6 The Parallel Input/Output Interface (PARIO) 64
7.7 The Time Stamping Bus (TSB) 64
7.8 The Field Programmable Gate Arrays 65
7.8.1 The Sample Rate Multiplier FPGA 66
7.8.2 The Time Base FPGA 67
7.9 Multiple DSMs in the Same System 69
7.10 Considerations for Production and Testing 70
7.11 Printed Circuit Board (PCB) Layout 70
7.12 Firmware for the DSM 71
7.12.1 Module DSM 72
7.12.2 Module RTC 72
7.12.3 Module SRM 73
7.12.4 Module MIX 73
7.12.5 Module DUART 74
7.12.6 Module GPS 74
7.12.7 Module MX4200 74
7.12.8 Module TESTAID 75
7.12.9 Message Passing 75
7.13 Conclusion 75

8 **Improvements in the Sample Pulse Generation and Time Stamping** 77
8.1 Introduction 77
8.2 Generation of the Sampling Clock Signal 77
8.2.1 SRM Modes 78
8.2.2 Sampling Modes 78
8.2.3 Sampling Control 79
8.2.4 Examples of Using the DSM Sampling Features 80
8.2.5 Timing Issues 81
8.2.5.1 Turning the Sampling On or Off 81
8.2.5.2 CADU Display Updates 81
8.3 FPGA Design 81
8.4 GPS Receiver Operation 82
8.5 System Time and Fundamental Frequency 82
8.6 Control Messages of the Standard DSM Application 82
8.7 Front Panel LED 83
8.8 Operating and Application Layer 84
8.8.1 Status Display 84
8.9 CADU Software Updates 85
8.9.1 CADU Date and Time Display (System Clock) 85
8.9.2 CADU Display of the DSM Status (DSM State) 85
8.10 Synchronised Measurements with CHART 86
8.11 Conclusion 87

9 Field Testing and Resulting Improvements **89**
9.1 Ripple Injection Measurements 89
9.1.1 Introduction 89
9.1.2 Instrument Setup 90
9.1.3 Measurements of the Ripple Control Signal 92
9.1.4 Results and Discussion 93
9.2 ELEC Feeder Measurements 94
9.2.1 Other Improvements to the CHART Software 95
9.3 The Getsync Program 96
9.4 The Chartdat Program 98
9.4.1 Analysing the Ripple Data With Chartdat — An Example 99
9.5 Connecting CHART to the Internet and Other Networking Aspects 99
9.6 Conclusions 102

10 Simulation of Quantisation Errors for Harmonic Analysis **103**
10.1 Introduction 103
10.2 Theoretical Considerations 104
10.3 Simulation 106
10.3.1 Model 106
10.3.2 Simulation Parameters 108
10.3.3 Results and Discussion 109
10.4 Discussion and Conclusions 111

11 Advances in Technology **117**
11.1 Processor Architectures 118
11.1.1 Power PC 119
11.1.2 Intel Pentium/Xeon and AMD Athlon/Opteron 120
11.1.3 DSPs 120
11.2 Benchmarks 121
11.3 Sample Timing Issues 125
11.4 System Architecture 126
11.4.1 Front-End Design 128
11.4.2 System Configuration Issues 130
11.5 Networking and Time Transfer 130
11.6 System Configurations and Third-Party Hardware 134
11.6.1 PC With Standard A/D Converter Card 134
11.6.2 Off-the-shelf Hardware 135
11.6.3 Custom-Built A/D Converter Card 136
11.6.4 Custom-Built Hardware 137
11.6.5 Other Hardware of Interest 137
11.7 Software Considerations 138
11.7.1 Operating Systems 138
11.7.2 Development Environments 139
11.7.3 Programming Languages and Application Software 140
11.7.4 Analysis Software 140
11.8 Commercial Data Acquisition Systems 141
11.9 Conclusion 143

12 Conclusions and Future Improvements 145
12.1 System Requirements 145
12.2 CHART III 146
12.3 Error Simulations 147
12.4 Advances in Technology 147
12.5 Future Enhancements 148

A Publications 151
A.1 Published 151
A.2 Prepared and Submitted 151
A.3 Application Note 153
A.4 AUPEC 2001 Paper 161
A.5 MELECON 2004 Paper 166
A.6 Sampling Rate and ADC Width Paper 170
A.7 System Requirements Paper 176
A.8 Implementation Considerations Paper 182
A.9 Microprocessor Advances Paper 188

B Chartdat Manual and Reference 193
B.1 Introduction 193
B.1.1 Invoking Chartdat 194
B.1.2 Help 194
B.2 Principal Operation of Chartdat 194
B.2.1 Return Codes and Error Messages 196
B.3 Time and Time Specification 196
B.4 Actions 197
B.5 Limits 198
B.6 Other Command Line Options 199
B.7 Examples 199
B.8 Future Improvements 200

C The DSM Schematic Diagrams 203
C.1 The DSM with TMS320C31 203
C.2 Connector P1 and P2 (TSB) 217

D The GAL Listings 219
D.1 MIX Address Decoder 219
D.2 DSP Address Decoder 220

E The FPGA Schematic Diagrams 223

F FPGA Configuration of the DSM 241

G Timing Diagrams of the Sampling Clock 245

H Photographs 249
H.1 CHART System 249
H.2 Ripple Injection Field Tests 255

References 259

List of Figures

2.1	Stability ranges of different oscillator types	7
2.2	The encoding of the DCF77 time signal station	10
3.1	Propagation of a line fault on an HVdc system	19
3.2	Typical waveform of a line fault	19
4.1	The maximum harmonic order which can be resolved	31
5.1	The maximum harmonic order which can be resolved (logarithmic plot)	43
6.1	CHART III system overview	52
6.2	A block diagram and overview of the CHART system	52
6.3	Data flow in the PPU	53
6.4	The path of the sample clock from the DSM to the DCM	56
6.5	Accumulation of time errors in a typical CHART system configuration	57
7.1	DSM data flow	60
7.2	DSM block diagram	60
7.3	MIX baseboard/ MIX module	61
7.4	Block diagram of the C31 DSP	62
7.5	Block diagram of the SRM FPGA	67
7.6	Block diagram of the time stamping FPGA	68
7.7	Operating more than one DSM in a system	69
7.8	Spacing of PLCC pins	71
7.9	Simplified flow diagram of the DSM firmware	72
9.1	Injection of the ripple signal into the distribution system	90
9.2	Synchronisation of two separate CHART units	91
9.3	Alignment of a data structure in memory	96
9.4	The help output of getsync	97
9.5	Block diagram of an internet connection for a PPU	100
9.6	Several CHART units networked with a CADU and a central processing host	101
10.1	Components of a harmonic analyser	104
10.2	The error vector of the quantisation	105
10.3	The trigonometry underlying the phase error Ep_k	105
10.4	Block diagram of the simulation model	107
10.5	The transfer function of the ADC model	107
10.6	Non-zero mean error magnitudes at very low harmonic amplitudes	109
10.7	Harmonic magnitude errors as a function of harmonic order	112
10.8	Harmonic phase errors as a function of harmonic order	113
10.9	Harmonic magnitude errors as a function of ADC width	114
10.10	Harmonic phase errors as a function of ADC width	115
10.11	Harmonic phase errors as a function of ADC width at different FFT lengths	116

11.1 A typical configuration of the CHART instrumentation system 127
11.2 A typical configuration of the proposed sampling hardware 127
11.3 Another possible configuration of the proposed sampling hardware 131
11.4 NTP peer timing 133
11.5 Using an external programmable pre-scaler for auto-ranging 135

B.1 The help output of chartdat 195
B.2 A sample display of the contents of a DEF file 201
B.3 Output produced by the "info" action 201
B.4 Output produced by the "view" action 201
B.5 Output produced by the "stats" action 202
B.6 Output produced by the "viewtimes" action 202
B.7 Output produced by the "fix" action 202

List of Tables

2.1	Accuracy and drift of frequency standards	7
2.2	Leap seconds from beginning to present	8
7.1	Contents of Multibus II interconnect EEPROM	61
7.2	The signals of the Time Stamping Bus (TSB)	64
7.3	The formats of the time stamp	64
7.4	Modes for loading the FPGA configuration bit pattern	65
7.5	DSM error codes displayed by the front panel LED	71
9.1	Time domain recording limits	94
9.2	Time domain recording limits with asynchronous transfers	95
10.1	Proportionalities for magnitude and phase errors	106
11.1	Execution times of a 128 point complex FFT on various processors	123
11.2	Execution times of complex FFTs with varying number of points, using fftw	124
C.1	Pinout of the P1 and P2 Multibus II backplane connectors	217
C.2	Jumpers used on the DSM	218
C.3	Memory map of the DSM	218
C.4	FPGA registers	218

List of Photographs

1	Kea, *Nestor notabilis*	v
2	The DSM board in the test rack, with attached test equipment	249
3	The DSM board in the test rack	250
4	The DCM with and without case	251
5	The RDCM in a ruggedised, weatherproof case	251
6	The DAPM board, with 3 F/O cables attached	252
7	A complete CHART unit	252
8	A CADU work screen	253
9	Laptop used as CADU	254
10	CHART set up at the Papanui substation	255
11	CHART set up at the Pages substation	256
12	An RDCM connected to the substation circuitry	256
13	The ripple injection equipment at Papanui substation	257
14	The ripple injection equipment at Papanui substation	257

Glossary

186/110B	A Micro Industries prototype board for the Multibus II, with a 80186 microcontroller.
320C31	TMS320C31. A fast 32-bit floating point digital signal processor from Texas Instruments [214, 215].
486/133SE	Processor board from Int$_e$l for Multibus II racks.
68HC16	A 16-bit microcontroller from Motorola [165–167] also equipped with some digital signal processing features.
68HC916	Same as →*68HC16* but with integrated flash memory.
80186	An Int$_e$l 16/8 bit microcontroller.
80486	An Int$_e$l 32 bit microprocessor.
8751	An Int$_e$l 8 bit microcontroller.
AC	Alternating Current.
AD	Analog Devices. Manufacturer of semiconductors in the USA. `http://www.analog.com/`
ADC	Analog-to-Digital Converter. An electronic circuit which converts analogue signals into a digital representation.
Ageing	(frequency ageing) Change in frequency due to internal changes in the oscillator [46]. (→*Drift*)
Alpha	64 bit CPU from DEC
AMD	Advanced Micro Devices. American manufacturer of semiconductors. `http://www.amd.com/`
ANSI	American National Standards Institute. One of the standards bodies of the USA. `http://www.ansi.org/`
API	Application Program (or Programming) Interface. A defined set of functions and their arguments for interfacing two software entities, e.g. application software with the operating system
ARM	Acorn Risc Machine. A RISC processor family with a comparatively small die size and low power consumption. `http://www.arm.com/`
AS	Australian Standard.
ASIC	Application-Specific Integrated Circuit. A chip containing custom-designed hardware. High one-off costs, but cheaper to produce than an →*FPGA*.
ASTRA	Commercial TV satellite(s) broadcasting over Europe.
AT&T	Manufacturer of semiconductors and fibre-optic components in the USA.
Athlon	→*AMD*'s version of →*Int$_e$l*'s →*Pentium* processor. 64 bit versions are now available.
BDM	Background Debugging Mode. A feature built into the →*MC68HC16* microcontroller to debug software. It allows absolute control over all microprocessor functions including those that are not available to normal application software (e.g. read-only

registers) using a 3-wire serial interface. A PC-based control and debug station can be connected with this interface.

BIH — Bureau International de l'Heure

BIPM — Bureau International des Poids et Mesures

BIST — Built-In Self-Test. Performed by a Multibus II Board at power-up.

BS — British Standard.

BSD — Berkeley's flavour of the Unix operating system.

Burr-Brown — Manufacturer of semiconductors in the USA.

C31 — Abbreviation for TMS320C31.

CADU — Control and Display Unit [127, 128]. An MS-DOS computer system connected to the →HUB to control the HUB and display data. The CADU is the user-interface of CHART.

CHART — Continuous Harmonic Analysis in Real Time [11,12,147,154]. Continuous Harmonic Analysis in Real Time. A general purpose real-time multi-channel data acquisition system suitable for measuring and analysing harmonics on power systems. Designed at the University of Canterbury.

Clock — Frequency standard plus counting mechanism [46].

CMC, PMC — Common Mezzanine Card; PCI Mezzanine Card [73, 74].

CORBA — Common Object Request Broker Architecture. Vendor-independent architecture and infrastructure that computer applications can use to work together over networks. `http://www.omg.org/`

CPU — Central Processing Unit. The microprocessor part of a computer which executes program instructions.

CT — Current transformer.

Cypress — Manufacturer of semiconductors in the USA. `http://www.cypress.com/`

DAC — Digital-to-Analogue Converter.

DAPM — Data Acquisition and Processing Module [149–152]. A MIX-module (→MIX) developed by the CHART project featuring 3 independent data processing channels equipped with a →*TMS320C31* →*DSP* each for processing the data. The analogue-to-digital convertors are connected to this via fibre-optic cables.

DAQ — Data Acquisition.

DC — Direct Current.

DCM — Data Capturing Module. Contains an ADC with input stage amplifier, and a fibre-optic interface. Data acquisition device of the CHART project. See →RDCM, →DAPM.

DFT — Discrete Fourier Transform. A mathematical operation (transformation from time to frequency domain).

DIN — Deutsches Institut für Normung e.V. (German institute for standards). Germany's (only) standards body. `http://www.din.de/`

DMA — Direct Memory Access. A technique to increase the performance of a computer where I/O devices access the computer's memory directly without using the CPU.

Drift — (frequency drift) Change of frequency with time, due to →*ageing* and other factors external to the oscillator [46].

DSM — Digital Services Module [102–105]. A MIX-module (→MIX) developed by the CHART project acting as a precision time base for data acquisition and event capturing.

DSP — Digital Signal Processor. A microprocessor with an instruction set optimised for typical signal processing tasks.

ECL	Emitter Coupled Logic. A technique for making logic circuits which uses bipolar transistors; it is very fast and very power consuming.
EEPROM	Electrically Erasable Programmable Read-Only Memory.
EMC	Electromagnetic Compatibility
EPROM	Erasable Programmable Read-Only Memory.
Ethernet	A standard local area network operating at 10 Mbit/s, 100 Mbit/s (Fast Ethernet) or 1 Gbit/s (Gigabit Ethernet).
F/O	Fibre-Optic.
FF	Flip-flop. A 1 bit storage element.
FFT	Fast Fourier Transform.
fftw	A very fast freely-available library for FFT computations [43, 45].
FIFO	First-In First-Out. Memory can be organised in this way.
FIR	Finite Impulse Response. A type of signal filter.
Firmware	Software for microcontrollers built into devices like hard disks or CD burners. Used to be in ROM, now it's in →*flash memory* so the manufacturers can fix their bugs after the product has reached the market.
Flash memory	Equivalent to →*EEPROM*. Large areas of the whole chip must be erased at once. Much cheaper than EEPROM an ubiquitous in consumer devices like digital cameras, or for storing firmware.
FP	Floating Point.
FPGA	Field Programmable Gate Array. This is a semi-application-specific integrated circuit.
G1, ..., G6	Motorola markets their PowerPC processor range for desktop computers under these names.
GAL	Generic Array Logic [117]. A kind of →*PLD* that uses →*EEPROM* cells as storage elements. It is electrically erasable and re-programmable.
Galileo	A satellite navigation system which is being built by the European Union. →*GLONASS*, →GPS.
GLONASS	Global Navigation Satellite System. A system of satellites allowing the determination of time and place anywhere on the Earth's surface. A project maintained by the government of the USSR. →*Galileo*, →GPS
GMT	Greenwich Mean Time. Time zone along the zero-meridian defined to have a time offset of 0. →*UT*
GNU	GNU is Not Unix (a recursive acronym). The world's first advocates of free software. The "free" refers to freedom as well as cost (otherwise known as free speech and free beer). `http://www.gnu.org/`
GPIB	General Purpose Interface Bus [78, 79]. Bus to connect laboratory instruments with each other. Originally developed by HP as HP-IB.
GPS	Global Positioning System. A system of satellites on low orbit covering all of the earth's surface. A project maintained by the US Department of Defence which can be used to precisely determine time and location. →*Galileo*, →*GLONASS*, →*SA*
GS	Geprüfte Sicherheit (tested safety). A seal used to identify electrical equipment that has passed this German electrical safety test.
HF	High Frequency.
Hitachi	Manufacturer of semiconductors in Japan.

HP	Hewlett Packard. Manufacturer of semiconductors and instrumentation equipment in the USA.
HSE	Harmonic State Estimation
HUB	The main processor card in the →PPU is referred to as "HUB" by CHART.
HVdc	High Voltage Direct Current.
I/O	Input/Output.
IC	Integrated Circuit.
IDE	1. Integrated Development Environment. Application software which combines editor, compiler and debugger for software development. 2. Integrated Drive Electronics. Interface used with hard disks where the control electronic is integrated into the disk, as opposed to being part of the computer.
IEEE	Institute of Electrical and Electronic Engineers. An organisation issuing technical standards and publishing many journals in the USA. `http://www.ieee.org/`
IEEE 1394	Standard defining a high-speed serial bus for interconnecting portable devices. Similar to →*USB*. Also known under the trademarks FireWire and i.Link.
Intel	Manufacturer of semiconductors in the USA. `http://www.intel.com/`
IP	Internet Protocol. The packet exchange protocol used by the internet.
IRIG-B	A definition of a collection of signal timings and encodings for transmitting time information over a serial line. The physical and electrical characteristics of the serial line are left undefined.
iRMX	Intel real-time multitasking operating system, running on Multibus II computers. Used by the CHART HUB/PPU.
ISO	International Organisation for Standardisation. `http://www.iso.org/`
ISR	Interrupt Service Routine. A special subroutine, or function, which is entered upon an external signal or internal condition, while normal program execution is suspended. When the ISR finishes, normal program execution is resumed.
JD	Julian Date. Number of days since the beginning of a 7980 year period. The first epoch is 1 Jan 4713 BC noon. Popular with astronomers because it is a simple number and time differences can be calculated easily. To make the numbers nicer and to align the day start with midnight, the modified Julian date of $MJD = JD - 2400000.5$ is commonly used.
LAN	Local Area Network. The network connecting computers within a room, building, or organisation.
LED	Light Emitting Diode.
LORAN-C	A radionavigation system.
LSB	Least Significant Bit.
MFLOPS	Million Floating point Operations Per Second.
mil	In connection with PCB layout this unit is often used as meaning 0.001 in.
MIPS	Million Instructions Per Second. Also a microprocessor family.
MIX	Modular Interface eXtension [82,84,89]. An Intel standard used for interfacing up to three hardware extension cards to a host processor. In this case, the host processor is located on the →MIX BB.
MIX BB	MIX Baseboard [88, 91]. A Multibus II card that is fitted with a MIX bus →MIX for interfacing extension cards.
MJD	Modified Julian Date. See →*Julian Date*.

Motorola — Manufacturer of semiconductors (among other things) in the USA. `http://www.motorola.com/`

MSB — Most Significant Bit.

MTBF — Mean Time Between Failure. A value expressing reliability.

Multibus II — A standard [69, 70, 80, 81, 83, 85–87] for a manufacturer-independent bus system. Up to 20 cards can be connected to a common back plane. This system is mainly supported by Intel.

NMEA — National Marine Electronics Association. GPS receivers generally use this standard for messages [171]. `http://www.nmea.org/`

NTP — Network Time Protocol. Commonly used internet protocol which keeps the clocks of computers in sync. Compensates for unpredictable delay times. Standardised by RFC2030 [157]. `http://www.eecis.udel.edu/~mills/ntp.html/`

OME 2 — On-board Module Extension. A standard from Siemens and Concurrent Technologies [201] for an I/O bus interfacing user-hardware to Multibus II systems. Modules are connected to a baseboard via a backplane on P2.

Opteron — A 64 bit version of the x86 architecture from →*AMD*. With multi-processor capabilities, comparatively low price.

OS, O/S — Operating System. Software which makes a computer boot, and handles basic I/O like keyboard input or disk file access.

OSS — Open Source Software. Software for which full source code is supplied. Beyond this, definitions vary and the term does not necessarily make a statement about the degree to which the licence permits redistribution, modifications, sale or use.

Oversampling — Sampling at a higher rate than required by Shannon's theorem. Usually given as a multiple.

P2 — The second of a 96-pin DIN connector which forms the →*Multibus II*. The signals on this bus are user-definable and they are used for the →*TSB*.

PARIO — Parallel I/O. An interface to the user on the DSM front panel, providing a number of digital I/O lines.

PC — Personal Computer. Originally the one from IBM, now probably means a computer with an Intel CPU (which may or may not be running a Microsoft operating system).

PC/104 — A bus system for embedded devices, using fixed-size PCBs ($90 \times 96\,mm^2$) stacked on top of each other. Electrically identical to the PC's ISA bus, PC/104-*Plus* additionally provides a PCI bus. `http://www.pc104.org/`

PCB — Printed Circuit Board. Flat piece of, often green, material connecting all the attached electronic components together.

PCI — Peripheral Component Interconnect. The bus system for expansion cards used in PCs. Bus speeds of 33 MHz and 66 MHz and widths of 32 bit and 64 bit are available. For embedded systems there is a 32 bit version with small connector, but identical electrical characteristic (CompactPCI). `http://www.pcisig.com/` and `http://www.picmg.org/compactpci.stm`

Pentium — Current versions of the x86 processor family from Intel.

PES — Power Engineering Society. A branch of the →*IEEE*.

PLCC — Plastic Leaded Chip Carrier. A common case for surface-mounted microchips where the pins have a J-like shape.

PLD — Programmable Logic Device. A device for implementing logical product terms in hardware. It contains an AND array followed by an OR array; the AND array is programmable.

PLL	Phase Locked Loop. A circuit which synchronises two frequencies by comparing their phase angles.
PMU	Phasor Measurement Unit
PowerPC PPC	A processor family from IBM and Motorola (→*G1*) with a →*RISC* architecture, which is commonly used in embedded devices and desktop PCs from Apple.
PPP	Point to Point Protocol. Running the internet protocol between two interfaces over a range of different media, including RS-232C.
PPU	Parallel Processing Unit [147, 154]. Main computer system of CHART.
PROM	Programmable Read Only Memory. A ROM which can be programmed once by burning away little connecting bridges on the chip.
PWM	Pulse Width Modulation. Often used as a simple method of digital-to-analog conversion.
PXI	PCI eXtensions for Instrumentation. Extends the PCI bus, used by PCs, to allow for the needs of instrumentation systems.
Python	A programming language for high-level scripting and rapid prototyping. `http://www.python.org/`
QoS	Quality of Service. Delivery of certain network packets within a defined time-frame. Used for e.g. internet telephony or audio/video streaming.
RDCM	Remote Data Capturing Module [30, 223]. A waterproof box with integrated power supply and battery backup containing a →*DCM*.
RFC	Request For Comment. The standards which describe how the internet and its services function are published like this. Any search engine will find them when searching for "RFC" and the number.
RISC	Reduced Instruction Set Computer. A microprocessor with a simple instruction set. The simplified architecture reduces design and programming complexity and power consumption, and allows a high execution speed.
RMS	Root Mean Square.
ROM	Read Only Memory.
RS-232C	The standard serial interface used e.g. by the IBM PC.
RS-422	A standard for a serial interface using differential data lines for high noise-immunity; the polarity between the two lines is reversed to represent the two binary digits.
RTC	Real Time Clock. A clock usually used as a time source.
RTOS	Real-Time Operating System. An operating system which guarantees a specified maximum execution time for its functions.
S&H	Sample and Hold. A circuit used with →*ADCs* which freezes the level of an analog signal for the duration of the digital conversion.
SA	Selective Availability. The accuracy of the →GPS can be deliberately reduced by some orders of magnitude.
SAR	Successive Approximation Register. A type of →*ADC*.
SBC	Single-Board Computer. A (usually relatively small) circuit board which contains everything needed for a stand-alone computer.
SCADA	Supervisory Control And Data Acquisition
SCSI	Small Computer System Interface. A parallel interface connecting peripheral devices to computer systems. Commonly used for hard disks, tape drives, etc. Standardised in ANSI X3.131-1986 (SCSI-1), and ANSI X3.131-198x, ISO/IEC 10288 (SCSI-2 proposal).

Semaphore A flag used to control access to a shared resource.

SGML Standard Generalized Markup Language. `http://www.w3.org/MarkUp/SGML/`

Siemens German manufacturer for a wide variety of electronic components, appliances, and industrial systems. `http://www.siemens.com/`

Sigma-delta Also delta-sigma. A type of →*ADC*.

SLIP Serial Line Internet Protocol. Running the internet protocol over serial lines, as opposed to over e.g. Ethernet. Now replaced by →*PPP*.

SRM Sample Rate Multiplier. Part of the →DSM which generates a sampling clock for the →RDCMS that is a multiple of the reference clock. The reference clock for harmonic analysis of power systems is their fundamental frequency.

SSH Secure Shell. A protocol or service for remote logins, which is encrypted and has further powerful features [207].

Stability (frequency stability) Change of frequency within a certain time interval [46].

TAI Temps Atomique International, International Atomic Time. This time scale is maintained by participating atomic clocks around the world. →*UTC*

TAXI Brand name for a chip set from →*AMD* that converts an 8-bit parallel microprocessor interface into a serial interface and back [4, 48, 162, 197]. Handles error detection/correction and control instructions. In CHART used for the fibre-optic interfaces between →RDCMS and →DAPMS.

TCP/IP Transmission Control Protocol/Internet Protocol. An internet communication protocol, possibly the most commonly used.

TCXO Temperature Compensated crystal Oscillator.

THD Total Harmonic Distortion. Square root of the sum of the squares of the harmonic magnitudes as ratio to the fundamental magnitude. $\frac{1}{A_1} \times \sqrt{\sum_{n=2}^{N} A_n}$

TI Texas Instruments. Manufacturer of semiconductors in the USA. `http://www.ti.com/`

TPNZ Trans Power New Zealand Ltd. Power distribution company in New Zealand and sponsor of CHART.

TSB Time Stamping Bus. A parallel bus used in →CHART to carry high-resolution time information from the →DSM to the →DAPMS. It uses the user-definable →*P2* bus of the Multibus II.

TTL Transistor Transistor Logic.

UPS Uninterrupted Power Supply. Unit which contains a battery and an inverter, to supply power when the mains supply has failed.

USB Universal Serial Bus. Commonly used for data transfer between portable storage devices. `http://www.usb.org/`

UT Temps Universel, Universal Time. One of the many time scales in use. Related to UTC. See page 8.

UTC Temps Universel Coordonné, Universal Coordinated Time. The time scale used in everyday life. It is based on UT1 and →*TAI*. See page 8.

VCO Voltage Controlled Oscillator. An oscillator with a frequency that can be trimmed around a centre frequency by a control voltage.

VDE VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V. (society for electro-technology, electronics, information technology). Governing body for electrical safety standards in Germany. `http://www.vde.de/`

VDI Verein der Ingenieure. The German society of engineers. `http://www.vdi.de/`

Virtual memory	The technique of using hard disk space when main memory (RAM) runs out. Memory which was least recently used is "swapped out" to disk, then used by another program. It causes an overall performance loss but has a number of advantages.
VLIW	Very Long Instruction Word. A technique of combining multiple instructions into one to increase throughput on parallelising microprocessors.
VME	VMEbus [65] (VERSAmodule Eurocard wasn't accepted as name). A bus or backplane system.
VT	Voltage Transformer.
VXI	VME bus Extensions for Instrumentation [66].
WAN	Wide Area Network. The network beyond the building or organisation.
Xeon	Intel's 64 bit version of the x86 architecture. Multi-processor capabilities.
XILINX	A company which manufactures →*FPGA*s. These are used by CHART. `http://www.xilinx.com/`
XML	Extensible Markup Language. `http://www.w3.org/XML/`, `http://www.xml.com/`

chapter 1

Introduction

1.1 Introduction

Monitoring of power quality is a significant requirement for power supply companies. One of the main factors affecting power quality is harmonic distortion. To maximise power transfer over distribution lines and minimise interference in communications, it is necessary to keep harmonic distortion and other disturbances to a minimum. To this end, legislation has been passed in some countries to limit the harmonics that customers are permitted to introduce into the distribution system. In order to limit harmonic disturbances, supply companies and their major customers need information about the voltage and current waveforms.

Excessive harmonics cause three primary problems in power networks:

- They result in an imaginary part to the power signal. This means that a proportion of the line carrying capacity is not available for carrying real power, which customers pay for.
- Therefore, equipment has to be built to higher specifications to carry the imaginary part of the power as well as sufficient real power.
- The imaginary power (current) causes line losses, which customers also do not pay for.

Within New Zealand, examples of significant harmonic problems include the influence of the Comalco NZ Ltd aluminium smelter at Tiwai Point, Southland, which draws a very large, mostly non-linear load. That single load consumes a high percentage of the total generator capacity in that area of the country. Another example occurred in recent years when one of the universities introduced "energy-saving" fluorescent light bulbs, which had such a high level of harmonic distortion that the power supply for the whole town was affected. On a smaller scale, commercial buildings with several hundred or more PCs also have issues with harmonic distortions because the PCs' switch-mode power supplies draw most of their current around the peaks of the sine wave, thus flattening the tops of the curves.

This thesis examines aspects of data acquisition for power systems, covering a range of applicatons from power quality monitoring to distributed comparative measurements. The focus is on instrumentation requirements.

1.2 Data Acquisition Systems Overview

Power distribution measurements are collected by an analog-to-digital data acquisition system. Such data acquisition systems are used in a wide range of applications. In general, they collect

analog data and store it for future analysis. Additionally, some systems may partially analyse the data before storing it. Systems can be classified by the number of data input channels available, together with the effective bandwidth and dynamic range of each channel. For a digital device, classification can be made by the data rate and resolution the input signal is quantised with.

Data acquisition systems are available commercially in a very wide range of configurations and are often optimised for a particular type of data. For example, a data logger for weather information may emphasise robustness and operation at extremes of temperature and humidity. Physical sturdiness, for example against animals[1], is a prime consideration. Such a device may, for example, collect values every 15 minutes, with a low requirement for timing accuracy (a difference of seconds, or even minutes, in the interval between samples is irrelevant). At the other end of the data acquisition system spectrum, large experiments in particle physics require well-synchronised short-term collection of huge amounts of data [13, 17, 195].

To monitor a 3-phase power distribution system, 6 analog input channels are required in order to measure voltages and currents of each phase. To monitor permanently, the system must be able to measure signals in real time and continuously without a break in the observed data. Sufficient processing power is necessary to repeatedly compute the Fourier transforms for a harmonic analysis, and to perform other numerical operations.

Therefore, in order to be suitable for power quality monitoring, a commercial data acquisition system requires:

1. At least 3 analog input channels. Each channel probably requires a separate ADC; multiplexing introduces too large an error in the sample timing unless the time required for all sequential samples is very much less than the sampling interval.
2. A maximum sampling frequency of at least 50 kHz (50th harmonic @ 50 Hz with 8× oversampling ≃ 50 kHz)
3. Continuous operation mode, or the ability to transfer data to a permanent storage device at a fast enough rate.
4. Time stamping of data, e.g. using an integrated GPS unit. The time stamping accuracy required limits the number of suitable commercially-available systems.

Data processing options programmed into the system, including FFT, are also very useful. Computing requirements can be reduced if the Fourier transforms are computed over full periods of the fundamental, which can be achieved by the sampling frequency being a multiple of the mains period.

1.3 The Requirement for Precise Synchronisation

Synchronisation and time stamping of samples is required whenever samples need to be taken at the same time at different locations. Synchronisation ensures that samples are taken at the same time; time stamping identifies the time at which each sample was taken, so the analysis software can compare them. For example, two very useful calculations for a power distribution network are the harmonic power flow and the harmonic state estimation (HSE). The harmonic power flow causes losses and takes up line capacity without supplying power to the consumer. A harmonic state estimation is a calculation of the harmonic content at each point in the distribution system, and is useful for tracing the origin of harmonic disturbances. Both of these calculations require analysis at different locations of a power distribution network simultaneously.

The synchronisation accuracy that must be achieved depends on the application, and can be in the order of 1 µs. For example, to be able to evaluate harmonic flows, the phase angles of all the

[1] The kea, a large New Zealand native mountain parrot (*Nestor notabilis*), has a well-deserved reputation for reconfiguring human-made hardware with its strong beak. (Page v has a photo.)

harmonic orders under investigation need to be compared between different sites. In general, the accuracy of this phase angle comparison should be in the order of a few degrees to a few tens of degrees to provide meaningful results. As a reference point, 1° of the 50th harmonic of 50/60 Hz correlates approximately to 1 µs.

1.4 Literature Overview

The literature relevant to this thesis falls into a number of broad categories:

1. Time keeping

   Methods of keeping time and their accuracy and stability are examined in section 2.1. This is the basis of any reference to points in time.

2. Time transfer

   Coordinating or comparing events in different locations is based on having the same time in each location. Time needs to be transferred from one place to another (section 2.2).

3. Synchronised data acquisition

   Collecting data simultaneously in different locations gives valuable information about phenomena if the data is collected at the same time, as discussed in section 2.3.

4. Power systems monitoring and quality

   The nature of power quality, and power quality applications for which synchronised data needs to be collected, are considered in chapter 3.

5. Accuracy requirements

   Different applications have different accuracy requirements, particularly for state estimation and therefore time stamping. These are summarised in section 3.5.

6. Digital signal processing and data acquisition

   The process of converting analog values into digital form introduces errors, which can be reduced by overall design of the system and digital filtering, as discussed in section 10.2.

7. Microprocessor technology evolution

   The evolution of more powerful microprocessors is having a major ongoing effect on data acquisition system design. Trends, features and examples of current equipment are reviewed in chapter 11.

1.5 Thesis Overview

Researchers at the University of Canterbury first identified the commercial potential of a system that performs such synchronised distributed power quality data acquisition in the late 1980s. An evaluation of existing commercial data acquisition systems found that none of them fully met all requirements. Most systems used multiplexed channels, which result in a lag between channels and a corresponding phase difference. Most of these systems could not handle the high data rate permanently, and only took a burst of samples and then analysed it, which does not fulfil the continuity requirement. No systems were able to time stamp samples with an accuracy in the microsecond range.

With the support of the national power distribution company (Electricorp New Zealand, later Transpower New Zealand), a research project was undertaken to develop a data acquisition system which integrates all the requirements for power systems harmonic monitoring. Its use was not restricted to this application only, it was designed to be a general purpose system with some added special

functions which made it particularly suited to power quality measurement. This thesis is concerned with the third generation implementation of this system, named CHART III. By this time it had progressed to the stage of limited commercial use.

The work described in this thesis can be divided into five parts. First, the next two chapters provide a review of the relevant literature. Because of the importance of data synchronisation to this work, this review begins with a brief history of time keeping, time transfer and existing synchronised data acquisition systems in chapter 2. Then aspects of power quality monitoring are considered in chapter 3.

This review of the literature is followed by a thorough discussion of the requirements for power quality monitoring systems. These requirements are presented in two parts: first, a discussion of the theory and its practical consequences (chapter 4), and secondly, a detailed ordered checklist of the requirements (chapter 5). The checklist effectively shows the complexity of such a system, and is designed to form a specification template for anyone who needs to design a system.

The next part of the thesis describes CHART III, a specific example of such a power quality monitoring system. CHART III is briefly described in chapter 6 to give background information. The author was particularly involved in the design of sample synchronisation aspects of CHART III. The time base is the key part in this synchronisation, and is described in detail in chapter 7. Field tests were undertaken to test synchronised data acquisition with this equipment, which are described in chapter 9. While the system successfully collected synchronised data, a number of shortcomings were identified, as can be expected with the first tests of a particular application for a prototype system. Chapter 9 ends with a description of how the most important of these issues were addressed.

These field tests were undertaken to gain experience with such a complex data acquisition system, and to investigate the issues involved in its deployment. As a result, a number of improvements were made to the time base, which are described in chapter 8. Furthermore, the experience was valuable for establishing the requirements such a system must meet for various applications, and is reflected in the requirements that are given in chapters 4 and 5. Because the emphasis of the field tests was on the instrumentation itself, no further analysis of the collected data is presented.

The fourth part of the thesis deals with a theoretical investigation of the errors inherent in a design like CHART III and the effect of varying the ADC width, harmonic order, harmonic amplitude and sampling rate. These simulations are presented in chapter 10.

Substantial improvements in general computing systems in recent years have had a major impact on the possible design of data acquisition systems, and aspects of the technology used in CHART III are now outdated. Therefore, the final part of the thesis, chapter 11, describes in detail options for a redesign of the system to current technological standards.

chapter 2

Time and Synchronised Data Acquisition

Synchronous distributed measurements require accurate time stamping, which in turn requires accurate time. Therefore a brief history of time keeping is presented below, followed by a discussion of how this time can be disseminated between data collection stations in a measurement system. The chapter is concluded by a review of a number of implementations that perform synchronised data acquisition, and issues associated with it.

2.1 Time Keeping and Time Scales

A physical reference to the entity of time can not be kept. So far, the standard of time has always been derived from a periodic event. A clock is built from periodic occurrences like the rotation of the earth, the swing of a pendulum, or the frequency of isotopic radiation, plus a counting mechanism. Counting periodic events does not by itself provide a framework for keeping time. In addition to counting, a method or system of making use of the counts is needed. This system is provided by a calendar, or time scale.

The "Western" or Christian calendar is based on years, months, and days and uses the earth rotation as basic measure for years and days. It starts with the year 1, prior events are determined by extrapolation. There is a good explanation of the Gregorian and Julian calendars, their history, adoption, and how the festive days are calculated in the Calendar FAQ [220].

Many other calendars are in use today or have been used in the past. For example, the Chinese calendar is a combination of solar and lunar period events and goes back to at least the 14th century BC. It is still used today for determining festive dates, but has been replaced by the Gregorian calendar otherwise [220].

The astronomical intervals on which calendars are usually based (earth rotation, moon orbit and earth orbit) do not divide evenly into each other. More or less complicated periods are needed to keep synchronisation to the astronomical reference, and are found in calendars as leap months or leap years. For example, the Thai lunar calendar adds a leap month approximately every three years, and the Persian Jalaali calendar presently adds a leap day to certain years within a 33 year cycle [18].

The day was undivided in Rome before the appearance of public clocks (ca. 250–150 BC). The division of the daytime and the night into twelve hours each originated in Babylonia, but is certain only since Alexander the Great (356–323 BC). Light and night time were independently divided,

and the length of these temporal hours changed with the season. Equal hours were known, but only used in the context of science and astronomy [225].

Sundials were in use since the 3rd millennium BC, probably originally only for establishing calendars (e.g. the solstices). The divisions of the day were added later. Earlier models produced only relatively inaccurate time of day readings. Hanging models for travellers could be adjusted for different latitudes[1].

For many centuries, the only way to measure the passage of time independently of weather and daylight were clepsydras, in which water drips out of a container through a small hole. These water clocks provided a more abstract measure of time (e.g. "a jug of time" in court), and their use for quantifiable observations goes back to the 3rd century BC. An elaborate mechanical machinery based on water power and telling the time in three different scales visibly and audibly was built in China in 1088 [232].

Precise date, location, and circumstances of the invention of the mechanical wheel clock are unknown. The oldest records in Europe date to around 1300. The critical new development was the use of an escapement for regulating the speed of the oscillation. Whereas the appearance of the escapement was not reflected in the language (the word for the mechanism remained "horologium"), the addition of an hourly bell-ringing mechanism around 1340 in Europe created a technical sensation and resulted in the use of the word "clock"[2]. The first designs with escapement used a verge and foliot (balance beam). Accuracy was much improved with the pendulum clock, patented by Christiaan Huygens in 1656. Later modifications included improvements to the escapement, and the addition of metal springs.

Mechanical watches and portable clocks could be made with an accuracy of a few seconds a month, making the seasonal variations of the length of the local solar day obvious to everyone. Public clocks were adjusted to the mean solar day instead, and public sundials removed (e.g. Berlin 1811). Railway traffic and telegraphy created a new urgency for common time references on a national and international level. Around 1900 the division of the world into 24 time zones and national dissemination of a common reference time were adopted [225].

Historic astronomical records [36, 130, 208] from Europe, China and the Arab world have been used to investigate the precision of the earth's rotation, and it was found that there is a long-term change in the length of the mean solar day, as well as other periodic components which exert an acceleration on the rate of rotation. Earth rotation is therefore not the best reference.

The most constant periodic event in nature found so far is the radiation emitted from certain isotopes when excited. Caesium, rubidium, and hydrogen are used in practice. The second as base unit of time became an international standard in 1889, and was redefined in 1967 to be

> The second is the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the ^{133}Cs atom.

in the Système International d'Unités (international system of units, SI).

Transitions between energy levels of the aforementioned atoms cause the emission of photons with a closely defined frequency. The "hyperfine" transition is an interaction between the magnetic dipoles of the electrons with the nucleus which occurs when the electron's magnetic N and S poles are induced to flip over [46].

The differences between atomic clock types are stability, cost, ageing and size. Hydrogen masers have a very high short and long-term stability, but also a high cost, size, and power consumption. Rubidium oscillators are small, comparatively light weight and cost effective, and a high performance model has a better short-term stability than a caesium oscillator from 10^0–10^4 seconds. Its major disadvantage is ageing. A caesium oscillator has a large size, weight, and a high cost and

[1] When the first sundial was taken to Rome in 263 BC, its incorrect display of the daylight hours for Rome because of a difference in latitude went unnoticed for 99 years [225].

[2] Related to the French *cloche* or German *Glocke*, bell.

Type	Frequency	Accuracy	1 s Drift in
Crystal (SiO_2)	wide range		1 day
Rubidium (Rb)	6.8 GHz		
Caesium (Cs)	9.2 GHz	10^{13}	>100 000 years
Hydrogen (H)	1.420 GHz		

Table 2.1: Accuracy and drift of frequency standards (from [46]).

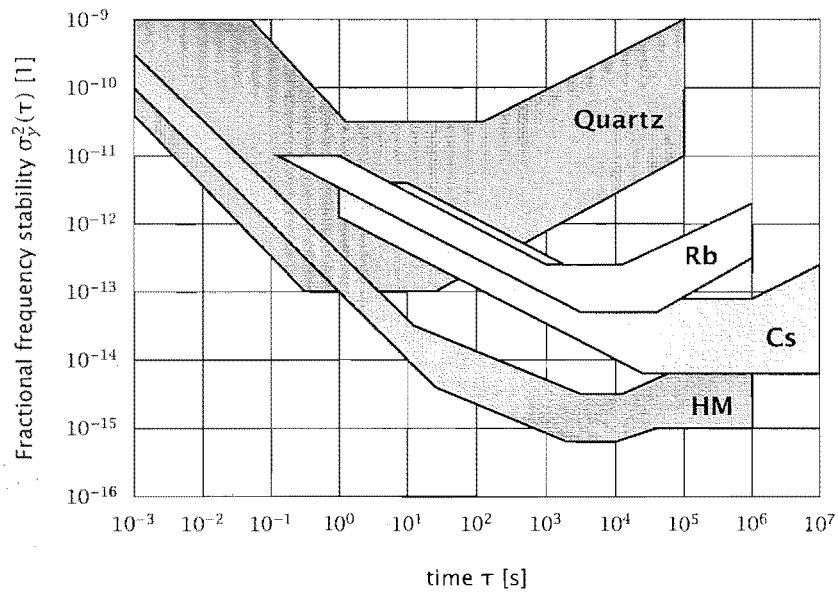


Figure 2.1: Stability ranges of different oscillator types (from [124]).

power consumption, but an excellent long-term stability (> 10^4 s). It has no ageing because it is a primary standard [46].

The most commonly used oscillator type is based on a quartz (SiO_2) crystal, and uses its piezo-electric property as a frequency reference. The construction is simple and reliable, cost-effective, small and has a light weight. The disadvantages are ageing and a high temperature coefficient, although the latter can be partially compensated for. Its accuracy is sufficient for most applications in civil life. The relative performance of caesium, rubidium, hydrogen and quartz oscillators are shown in table 2.1 and figure 2.1.

The following time scales are commonly used, or commonly referred to:

UT0 Temps Universel 0, Universal Time 0.
Mean solar time of the zero meridian obtained from direct astronomical observation.

UT1 Temps Universel 1, Universal Time 1.
UT0 corrected for the effects of small movements of the earth relative to the axis of rotation (polar variation).

UT2 Temps Universel 2, Universal Time 2.
UT1 corrected for small seasonal fluctuations in the rate of the rotation of the earth.

TAI Temps Atomique International, International Atomic Time.
The time reference coordinate established by the BIPM (Bureau International

Gregorian	Julian (JD)	Modified Julian (MJD)	TAI−UTC	GPS−UTC
1972-01-01 00:00:00	2441317.5	41317	10 s	
1972-07-01 00:00:00	2441499.5	41499	11 s	
1973-01-01 00:00:00	2441683.5	41683	12 s	
1974-01-01 00:00:00	2442048.5	42048	13 s	
1975-01-01 00:00:00	2442413.5	42413	14 s	
1976-01-01 00:00:00	2442778.5	42778	15 s	
1977-01-01 00:00:00	2443144.5	43144	16 s	
1978-01-01 00:00:00	2443509.5	43509	17 s	
1979-01-01 00:00:00	2443874.5	43874	18 s	
1980-01-01 00:00:00	2444239.5	44239	19 s	
1980-01-06 00:00:00	2444244.5	44244	19 s	0 s
1981-07-01 00:00:00	2444786.5	44786	20 s	1 s
1982-07-01 00:00:00	2445151.5	45151	21 s	2 s
1983-07-01 00:00:00	2445516.5	45516	22 s	3 s
1985-07-01 00:00:00	2446247.5	46247	23 s	4 s
1988-01-01 00:00:00	2447161.5	47161	24 s	5 s
1990-01-01 00:00:00	2447892.5	47892	25 s	6 s
1991-01-01 00:00:00	2448257.5	48257	26 s	7 s
1992-07-01 00:00:00	2448804.5	48804	27 s	8 s
1993-07-01 00:00:00	2449169.5	49169	28 s	9 s
1994-07-01 00:00:00	2449534.5	49534	29 s	10 s
1996-01-01 00:00:00	2450083.5	50083	30 s	11 s
1997-07-01 00:00:00	2450630.5	50630	31 s	12 s
1999-01-01 00:00:00	2451179.5	51179	32 s	13 s
2006-01-01 00:00:00	2453736.5	53736	33 s	14 s

Table 2.2: Leap seconds from beginning to present. The first leap second in 1972 caused TAI to be 10 s ahead of UTC. So far all leap seconds have been positive, i.e. inserted. The GPS epoch is 00:00:00 UT 6 Jan 1980.

des Poids et Mesures) on the basis of the readings of atomic clocks operating in various establishments in accordance with the definition of the second.

UTC Temps Universel Coordonné, Coordinated Universal Time.
Corresponds to TAI but offset by an integral number of seconds. UTC is adjusted by inserting or removing leap seconds so that the difference to UT1 is less than 0.9 s.

GMT Greenwich Mean Time.
The National Physics Laboratory, UK states [116] that authorities are not agreed on whether GMT equates with UT0 or UT1, however the difference would be in the order of milliseconds. GMT is no longer used for scientific purposes, and is the basis of civil time for the UK.

Of practical relevance are TAI and UTC. TAI is a primary time scale derived from atomic clocks located all over the world by calculating a weighted average. It is a deferred-time time scale and readings are only available after a few weeks, with a best-case accuracy of 10-20 ns [49]. UTC is adopted by most countries as their official legal time, with a constant time difference of time zone and daylight saving applicable to each country. The issue of the leap second is somewhat controversial, but so far the advantages are deemed to outweigh the disadvantages [169]. The leap seconds applied so far, and the resulting differences between TAI and UTC, are shown in table 2.2. It is necessary to distinguish between a definition of a time scale and its realisation. In the case of

TAI, the particular realisation is given by the name of the laboratory in the subscript. Guinot [49] gives a superb overview of the time scales in use and their relations and definitions.

From the point of view of synchronising measurements for power quality monitoring, the relevant factor is that all participating instruments are synchronised to each other with sufficient accuracy. The choice of time scale is of secondary importance, as long as the same one is used for each instrument.

2.2 Time Transfer

Time transfer is the transportation or dissemination of time information from one location to another. Communications connections are frequently used for this: cables (copper or optical), or electromagnetic waves (all frequencies are used). For synchronous measurements, all involved instruments must have a common time reference, that is, the area of time transfer must cover at least the area in which measurements are undertaken. Whether the time transfer happens from one (primary) instrument to the other instruments, or from an independent time source to all instruments, is of no consequence.

For comparative distributed measurements, all involved instruments need to be synchronised with each other. A synchronisation with a global time scale like UTC or TAI is not strictly necessary, unless a known temporal relationship to absolute time is desirable for other reasons. Because power grids are usually confined to continents and do not cross large bodies of water, power quality measurements are performed on a regional basis. Global synchronisation is therefore not a requirement.

Time signals can be transferred over long distances at extremely high accuracy using dedicated optical cables [99]. However, this approach is not practical in most situations because of the expense and difficulty of cable installation. It also greatly reduces the flexibility and portability of the instrumentation systems, therefore it is not discussed further. A number of other options exist which allow the transfer of time signals at an accuracy adequate for power quality monitoring, which are discussed in this section.

The obvious means of disseminating time to a wider area is a radio transmitter. Many countries were and are operating these terrestrial time signal stations to disseminate their national official time [100, 131]. The operational range of the time transfer is equivalent to the reception area of the transmitter. Receivers can be made relatively simply and inexpensively. A major advantage of this method is that reception inside buildings is unimpaired because of the very low or low frequency usually used. A clock based on the time signal of one of these stations is commonly referred to as a radio clock.

The time information must somehow be encoded into the carrier. For example, figure 2.2 shows the encoding scheme for a 77.5 kHz transmitter near Frankfurt[3] which broadcasts UTC_{PTB} [54, 183]. The carrier amplitude is reduced to 25% at the start of each second, for either 100 ms or 200 ms, transmitting one bit of information. This reduction is skipped in second 59 to mark the end of the minute, which explains the peculiar behaviour of German railway station clocks where the second hand remains on 58 for two seconds before advancing directly to 60. Receivers for this are very inexpensive at a starting cost of about NZ$5 in an electronics shop. Radio alarm clocks are not much more[4]. The carrier frequency itself is a standard and can e.g. be used for calibration [209]. The achievable accuracy is in the order of a few milliseconds, but can be substantially improved with signal processing techniques and the use of the pseudo-random binary phase shift keying (BPSK) of the carrier [228]. Lichtenecker [131] reports the accuracy as being 10–50 ms without BPSK and 10–50 µs with BPSK.

Television transmitters are often simultaneously used as frequency reference, by linking the carrier frequency to a frequency standard. This does not transfer time information by itself, but can be used to keep a local clock accurate after an initial time transfer performed by other means.

[3] Mainflingen, 50° 01' N, 09° 00' E, approx. 25 km South-East of Frankfurt/Main.

[4] The author built one of these radio clocks from a commercial kitset in 1984.

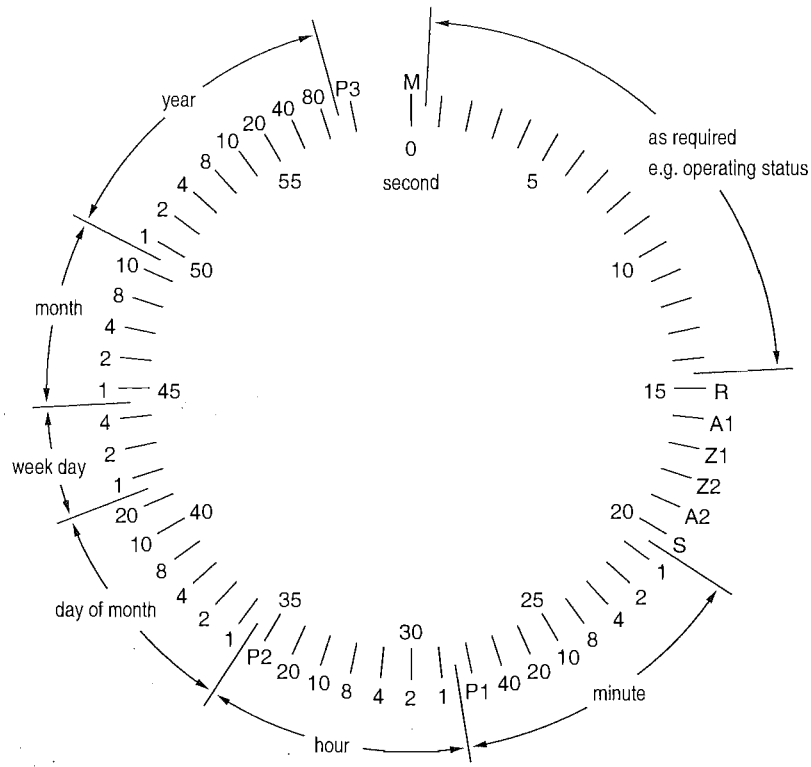


Figure 2.2: The encoding of the DCF77 time signal station in Mainflingen near Frankfurt, Germany. Time and other information is transmitted over a period of one minute at a rate of 1 bit per second [183]. Digits are BCD encoded. R: currently using the backup-antenna; A1: advance notice of daylight saving change; A2: advance notice of a leap second; P1, P2, P3: checksum bits.

Another possibility is to use the teletext signal for time transfer, at least in those countries in which it is available. A simple interface to the teletext receiver chips which are used in consumer electronics is sufficient [118]. Using the European ASTRA satellites, a comparison of the output of two receivers tens of metres apart shows a precision of ± 45 ns after calibration at the same position to compensate for cable lengths. The long term (6 months) drift was < 5 ns. The authors expect to achieve 25 ns accuracy (standard deviation) over a distance of 1000 km when taking satellite position into account. This accuracy would certainly be sufficient for comparative power quality measurements. The exact relationship to time scales like UTC or TAI is not discussed. This solution potentially provides a large degree of geo-political independence.

Terrestrial radio-navigation systems can also be used for synchronisation and time transfer. As with TV signals, their service area is limited to within a maximum distance from the transmitter. For example, the LORAN-C system operating in the North-West Pacific is reported to achieve a time transfer accuracy of 0.1–0.2 µs [5] or 3.6 µs [56].

Time signals are available via telephone modem from a number of European countries using the "European telephone time code". This system can measure the round-loop delay and correct for the one-way delay. It results in uncertainties in the transferred time of a few milliseconds, and no systematic errors were observed over the measurement period of 100 days [55]. The equipment used for this is portable and can be deployed anywhere to synchronise regional instruments to a single clock.

The internet is a widely established network for data transport, and has been used for time transfer for over 20 years. The main problem for time transfer is that the internet is a packet-switched network with indeterminate delays. The most widely used protocol is NTP / SNTP, (Simple) Network Time Protocol [157–161]. The protocol was designed to perform over a network with unknown and changeable delays, and compensates for the network's timing characteristics as much as possible.

It is a client/server architecture, and each NTP node by default acts as both server and client. The long-term accuracy is identical to that of the clock which is used as primary time source, and is likely to be close to TAI (international atomic time), as time servers which themselves are linked to TAI are readily available all over the internet.

With a new implementation of an NTP-client [33] it was found that an accuracy of 1 ms (the minimum java can resolve) can be achieved with loopback networking (i.e. to the same host). This acts as a sanity check of the algorithm, as synchronising a host to itself is of little practical value. For time transfers over physical networks, the accuracy is limited by the network layer delays. Factors which contribute to the network layer delay of a packet from host A to host B are the delay before a packet leaves the host, the propagation delay over the medium, the time needed to process the packet in intermediate routers, and the time for which the packet waits in a queue of intermediate routers [93]. The times other than the queueing delay are fairly determinate. For an idle network, the queueing delay is practically zero, for a congested network, it becomes the dominant contributing factor. Any algorithm for time transfer over the internet will have to take into account the possibility that any packet can be simply discarded e.g. in an overload situation.

It should be pointed out that time transfer does not have entirely the same requirements as information transfer. While some applications may require the transfer of information in the shortest possible time, transfer delays do not degrade time transfer accuracy significantly as long as the delays are deterministic and are allowed for. Also, time transfer algorithms are robust against a certain percentage of packet loss.

For time-critical transfer of protection information, IP (internet) networking is deemed to be sufficient as long as a separate non-IP based method for highly accurate time synchronisation is deployed as well [198].

The accuracy which can be achieved over packet-switched networks depends on several factors, including the accuracy of the reference clock (of the time server), the delay characteristics of the network, and the algorithms used. A number of different algorithms were developed for different needs. Using an NTP server and a new algorithm for the client, an accuracy of 2 µs rms can be achieved over 1200 km [120]. An improved version [121] which does not use NTP but allows an explicit tradeoff between accuracy and cost (network traffic, CPU cycles) can achieve a maximum accuracy of 8–30 µs rms. No statements were made about the level to which the networks were utilised by other traffic.

Reported uncertainties for NTP vary considerably and tend to be lower with recent computer hardware and networks. Over long distance 1–10 ms can be achieved [194], and better than 1 ms over a LAN [161]. Integrating the NTP protocol with the Ethernet can yield dramatic improvements; a higher degree of integration results in larger improvements. The limits for this are approaching 1 µs, but this effectively amounts to hardware time stamping [202]. A more detailed discussion of the limits of current technology for NTP time transfers can be found in section 11.5.

It is a non-trivial exercise to design a highly accurate clock which satisfies requirements for both internal synchronisation (between clocks of a distributed system) and external synchronisation (between the system and external time scales, etc. UTC). The elaborate design by Schossmaier et al [195] is counter-based and implemented in an ASIC (application specific integrated circuit), and provides a certain amount of redundancy and self-test features as well as time stamping. A clock of this design can be used at each site involved in distributed measurements, but whether this is in excess of the requirements of the particular power systems application should be investigated.

Satellites provide a convenient relay point for radio waves over long distances. "Common view" transfers are often used, in which both sender and receiver have direct line-of-sight of the same satellite. A large body of literature exists, e.g. [124–126, 144]. Common view transfers are effective because the distance which the signal has to travel is constant (neglecting slight variations in satellite positions) and largely independent of weather conditions. Radio signals which rely on being reflected back to Earth by layers of the atmosphere have a strong dependency on weather conditions and large variations in pathway and distance.

2.2.1 Satellite-Based Navigation Systems

Satellite systems which were established for navigation can also be used for time recovery, because determining precise position through signal propagation delays is closely linked to determining precise time. Two such systems exist, the US-American Global Positioning System (GPS) and the Russian Global Navigation Satellite System (GLONASS), both covering all of the Earth. Both these systems are approximately identical in performance [37] and allow time recovery with submicrosecond accuracy. The European Galileo system is in the process of being deployed. A large volume of publications exist for both GPS and GLONASS.

Levine [122] gives a detailed discussion of time transfers with GPS, GLONASS and Galileo, and two-way satellite transfers. The transfer process itself can be achieved with a fractional frequency uncertainty of less than 10^{-15}. The state of technology in 1999 allowed the use of GPS as a time source with an accuracy of 10–25 ns [123], which should satisfy the most stringent power quality application.

The Global Positioning System is comprised of 4 evenly spaced satellites on each of 6 planes, orbiting the earth at a low altitude of 20 183 km every 12 h. The satellite constellation is optimised to make the highest number of satellites visible at any location on earth. This makes the system redundant to a certain degree against satellite failure, satellites obscured by buildings, etc. and problems with receiving particular satellites. The receiver can be moving or stationary. Each satellite continuously transmits a unique signal on the two frequencies L_1 = 1 575.42 MHz and L_2 = 1 227.6 MHz.

These signals carry enough data for computing navigational positions and their accuracy. For an accurate calculation 4 satellites must be visible. After determining the position of the receiver, accurate time recovery is possible by tracking only 1 satellite. It is necessary to have line-of-sight to the satellites when positioning the receiver's antenna. Galileo will provide an advantage in this regard once it is available, because it is designed to also work inside buildings.

Since 1990, GPS time is given by a composite of base station and satellite clocks and is steered to within 1 µs of UTC_{USNO}, neglecting an offset of an integral number of seconds. Because GPS time is not adjusted for leap seconds, there is a varying number of seconds difference between GPS and UTC. The actually applicable difference is broadcast by the GPS and most likely already applied by the receiver. (This information is readily available from the United States Naval Observatory.) The relationships between the time scales TAI, UTC and GPS are shown in table 2.2 with all historic leap seconds so far.

Commercial GPS receivers provide a 1-pulse-per-second signal, `1pps`, which is synchronised to the beginning of each second to within the accuracy of the time recovery. The information to which second each `1pps` belongs is usually available from the receiver via a serial interface, which is also used to set up and control the receiver.

The accuracy of the GPS may be artificially degraded by added pseudo-noise. Comparisons with a rubidium atomic clock (which has excellent short-term stability) have shown [182] that this pseudo-noise has a maximum amplitude of ±300 ns. Programming GPS receivers to use the same satellite for time recovery ensures that the artificial error is identical for both receivers and therefore does not prevent sampling synchronisation, although the error appears as phase noise. If the particular application does not require an accuracy better than this, the pseudo-noise can be ignored. Schoukens [196] gives expressions for the measurement errors for sinusoidal phase noise caused by clock jitter.

Operating two GPS receivers located closely together in "differential mode" yields a relative distance accuracy in the order of centimetres. Because it does not improve time synchronisation between measurement sites, it is not relevant for distributed measurements.

2.3 Synchronised Data Acquisition

Synchronisation allows data points which are obtained at different nodes of the measurement setup to be meaningfully related to each other. Nodes, in this context, can range from a set of cards in the same PC, to instruments in a large hall, to instruments thousands of kilometres apart. Synchronisation can be achieved by designating one of the nodes as master, which controls the sample timing of all the slave nodes. This is typically used for nodes located in the same instrument. When multiple instruments are involved, synchronisation between master and slaves can be achieved with interconnecting cables, but cable propagation delays need to be taken into account. For large distances, direct links between instruments are impractical, and synchronisation to an external clock via time transfer to all instruments is more feasible.

Precisely synchronised data acquisition is required for applications across a wide range of scientific disciplines. In power systems, synchronisation is used for example in phasor measurements, fault location, mutual line inductance measurements, and adaptive relaying, monitoring and control. Outside the field of power systems, applications include astronomical measurements, plasma fusion experiments, real-time operating systems and cellular telephony. This section first describes several examples of such data acquisition systems, and their architectural design and key features. Then other issues with synchronised data acquisition are discussed.

A data acquisition system designed for collecting astronomical information over an area with 1000 km diameter on the Earth surface is described by Pryke and Lloyd-Evans [181, 182]. The system requires a large number of nodes and is specifically intended to time stamp data at the microsecond level while having as low a per-node price as possible. Therefore, relatively low-cost GPS receivers are used, combined with custom-built time-tagging electronics, and PCs. Tests showed that these receivers provide sufficient accuracy for the experiment. The tests used two co-located receivers, 500 m apart, and a cable connection with known (measured) delay. With each receiver using the same satellite, a fixed offset between them of $\ll 10$ ns was observed, and a random error with a standard deviation of ≈ 6 ns. The tests were repeated over a baseline of 11 km using a microwave (L-band) link. The relative error between the receivers had a standard deviation of ≈ 7 ns. The absolute offset is unknown because the link delay was unknown, but unless extreme weather conditions prevailed, the link delay varied by < 1 ns/day.

A very extensive data acquisition system has been purposely built for a plasma fusion experiment [15, 203–205] by EURATOM. Traditionally data had been analysed after the experiment finished, but an increase in data volume and a requirement for closed-loop control and real-time monitoring for safety reasons necessitates data analysis during the experiment. The physical size of the experiment and the number of measurement points make a single-enclosure design impossible. A distributed design was chosen with nodes connected with fast bi-directional links in a tree-type topology. The system can be extended to approximately 250 major nodes up to 300 m apart from each other. Each major node can accommodate a number of event and analogue inputs, as well as generate a number of output variables used to control the experiment itself. Data acquisition and control of the experiment are merged, allowing real-time control depending on pre-defined conditions. A fast, prioritising, packet switching network using the Infiniband standard allows each node to exchange event information with any other node. Because the relation to external time is of little relevance but the precise synchronisation within the experiment is of utmost importance, all nodes are slaves to a central timing unit. The synchronisation of sampling, event handling and time stamping is via synchronous 960 MBaud duplex optical 62.5/125 µm fibre connections with 850 nm wavelength. The task of real-time data processing is handled by a combination of DSPs and FPGAs. The FPGAs are reprogrammed for particular tasks, which adds flexibility.

Mutual inductance of power transmission lines can be measured relatively easily while all lines are out of service, however taking lines out of service is undesirable and not always possible. The mutual inductance can be calculated from the measured voltages and currents of all phases at each end while a large step in zero-sequence current occurs, e.g. by briefly disconnecting one phase. If voltages and phases are measured at each end with GPS synchronisation, the mutual line inductance can be calculated [233] for a live system.

Cellular telephony can only function correctly when the time in cells is closely synchronised. Each cell contains a transmitter/receiver and serves phones within a certain area. A large volume of publications exist for achieving cell synchronisation by a variety of different means. To increase redundancy and robustness, strategies exist for synchronisation when an externally supplied time (such as e.g. GPS) is not available [1], by utilising the communication channels which exist between cells. The data communication channels between distributed measurement instruments can be used in a similar manner, as far as they exist, but the techniques developed for time transfer over packet-switched networks are likely to be more appropriate.

A high precision interrupt or event timer designed for use by distributed real-time operating systems [50] applies equally to distributed measurement systems. The GPS is used as basic time source, and the time stamping and event queueing is implemented in an ASIC (application specific chip) to keep the burden away from the processor and to achieve high accuracy.

The type of analogue to digital converter (ADC) has an influence on the overall system design and performance. Dual-slope integrating converters are not suitable for instrumentation systems because of their very low sampling rates. Flash converters have a limited resolution but can operate at extreme frequencies. They may be considered for transient recording, but successive-approximation converters should be adequate for that, too. Successive-approximation converters are relatively easy to use and low cost. Delta-sigma converters can have a high resolution, do not have a differential non-linearity error [206] and are low cost. They are essentially a 1-bit converter using oversampling and an internal digital filter. Their major disadvantage for synchronised sampling is that the internal digital filter introduces an unknown group delay and that the conversion clock is a multiple of the sampling clock, which causes loss of the relationship between the timing of the sampling and the sample data appearing.

The advantages of oversampling intrinsic in a delta-sigma converter are not by themselves a reason to choose this type of ADC, because a system based on a successive-approximation converter can also be designed using oversampling, thus providing the same advantages. Oversampling pushes the roll-off frequency of the input analog low pass filter up, simplifying its design and error contribution. The benefits of oversampling on the digital processing are discussed in chapter 10.

Phasor measurements are a common power systems application of synchronised data acquisition. Knowledge of phasors at various parts in a system and their mutual relationships e.g. allows the calculation of a state estimation, instability prediction, or implementation of adaptive relaying and control [174]. A number of techniques exist for achieving this synchronisation, e.g. microwave or optical fibre links, or AM radio or satellite broadcasts [175]. Measuring the time difference between the sampling times and deriving a correction factor from it [115], or correcting local clocks for the delay of a common synchronisation signal [173], yields coherent phasor information. Using the GPS for synchronisation has become common practice and is now the most economic option, which also provides an absolute time reference. Comparing phasors is easiest when they are tagged with an absolute time, which is stipulated by the IEEE synchrophasor standard [71].

Traditionally, phasor timing has been established by synchronising the sampling clock to an external timed signal (e.g. `1pps` output of GPS receiver). It is possible to establish the time relationship of a phasor to UTC in the frequency domain when using a free-running fixed sampling rate [206]. A 20 MHz counter in the processor and a GPS `1pps` connected to an interrupt input of the processor are used to subdivide the one second intervals. A compensation is then applied to the phasor by adding an appropriate phase shift. Simulations confirm that this works [129]. The authors do not discuss how well this method works for the measurement of harmonic phases for the purposes of harmonic power flow calculations.

The time stamping system described by van As [224] is implemented in hardware to achieve accuracies in the 1 µs area. It uses a real-time clock which is synchronised via the GPS. The `1pps` signal of the GPS receiver is subdivided using counters. The implementation is similar to the one used for CHART III (described in chapter 7). The system is generally useful for wide area measurements and is applied to fault location using the travelling waves method.

The GPS can be used for the generation of synchronised signals as well as for synchronising sampling clocks. Superimposing audio-frequency signals on the power system's fundamental is commonly used for controlling e.g. appliances or street lighting. All audio-frequency generators need

to be synchronised, which is commonly achieved using a phase-locked transmission of the signal from a master control point. Alternatively, decentralised generation of these ripple control signals is possible when using the GPS to synchronise their phases [145].

chapter 3

Review of Power Quality Monitoring

3.1 Introduction

The previous chapter ended with a discussion of synchronous data acquisition systems in the general sense; this chapter discusses their application to power systems monitoring.

Synchronous measurements are the basic ingredient of a range of potential applications when applied to power systems [174]: measuring voltage and current phasors, state estimation, instability prediction, adaptive relaying, and control.

GPS synchronisation is reliable and accurate enough for most applications [185]. The measurement of a disturbance of a real system was found to match the simulated behaviour of the system.

Another important issue is that of harmonic power flow. This is undesirable, because the distribution network needs to be dimensioned to carry the harmonic power as well as the fundamental frequency active and reactive power. This harmonic power is dissipated as heat in the system. Phasor measurements of voltage and current provide a picture of harmonic power flows on a distribution network.

Harmonic state estimation is the determination of the harmonic power flows in a distribution network. Time-synchronous measurements are required at multiple nodes in the network, and all of the obtained data together is required for the calculation. To compute the HSE completely, it is not necessary to measure voltage and current magnitudes and phases at every node in the distribution network. It has been shown [41] that a minimum number of points exist from which the harmonic state of the whole system can be calculated. The internet can be used to communicate the measurement data [2], at least as long as instantaneous results are not required. Using the internet introduces an unpredictable time delay for data transport.

3.2 Phasor Measurements

A phasor is a vector representation of a sinusoid in a polar coordinate system. The length of the vector corresponds to the sinusoidal amplitude, and its angle corresponds to the angular position within the sinusoidal period at that particular point in time. The phasor rotates by a full circle for each sinusoidal period. On a transmission line, line losses would be represented by a reduction of the length of the phasors observed at the beginning and end of the line. A difference in fundamental phase angle between the ends of the line is indicated by an angular difference between the two phasors. Phasors are determined independently for voltages and currents, and are a convenient method for calculations related to power systems.

Comparing phasors obtained from different locations of a power system implies that their measurement is synchronised (or that the phasor data is time stamped with high accuracy to a common time reference). Potential applications of synchronised phasor measurements include [188] state estimation, fault location, system control, and event recording. Synchronisation and/or time stamping can be performed with the GPS [42, 186, 188]. The use of the internet is becoming increasingly common to bring the data from multiple phasor measurement units together for analysis [188].

Phasor measurement units which are GPS-synchronised have been used for harmonic measurements and disturbance monitoring in New York state [42], and data has been analysed in real time. A GPS accuracy of $\pm 0.5\,\mu s$ allows determination of the phase of the fundamental to $< 0.02°$, and of the 25th harmonic to $< 1°$.

Another implementation of a phasor measurement unit [186] measures voltages and currents with $< 1\,\mu s$ accuracy. It uses a sigma-delta ADC, a sampling rate of 2.88 kHz, and generates 12 points per fundamental period which are processed by a recursive discrete Fourier transform. It is used for fault recording, disturbance recording, and verification of generation and transmission modelling.

Phasor measurement units can exchange data in a standardised format [71] which includes time stamps with the data.

3.3 Power Quality

The term "power quality" covers a range of electricity supply disturbances. These include dips or sags (supply voltage below nominal value), flicker (observable brightness changes in lighting, caused by fluctuating supply voltage), brownouts (deliberate voltage reduction to avoid a blackout), substantial or complete supply failure for $\ll 1$ s, spikes (very short substantial rise of supply voltage to several times above its nominal value), and harmonic distortions. All these disturbances vary with time.

Many possibilities exist for the causes of these disturbances. Spikes are often caused by electrical machines and their thyristor controls. Harmonic distortions are usually caused by non-linear loads. Fluorescent lamps with their auxiliary circuitry, and computer power supplies, can both cause localised harmonic distortions. A larger problem for the supply industry however are large non-linear industrial loads, which can influence the supply network over large distances. In New Zealand, this is especially true for the South Island, where the Tiwai aluminium smelter and the HVdc link represent huge loads in a relatively small electrical system. Harmonic power monitors are used to investigate this problem.

Flicker has been investigated by Keppler as PhD research [94], and in subsequent publications by Keppler and others [95–97].

Power systems monitors have traditionally been built using a digital signal processor for handling the numeric computations, perhaps combined with a standard processor handling the remaining tasks. Advances in microprocessor technology and their improved numerical capabilities make simpler designs possible where the signal processor is no longer required. A system for analysing sag on multiple channels using a standard desktop PC microprocessor has been developed [32].

The use of current microcomputer technology in power systems is discussed in detail in chapter 11, including further literature pertaining to the subject.

A general trend towards server-based applications can be observed over recent years for anything which is, or can be, related to the internet. Power quality applications are no exception and efforts are being made to offer web-based services [31]. All the processing is moved to a server, and control is via the client's web browser (browser user interface). Communications between parts of a system, like data acquisition units and processing units, which are based on open standards

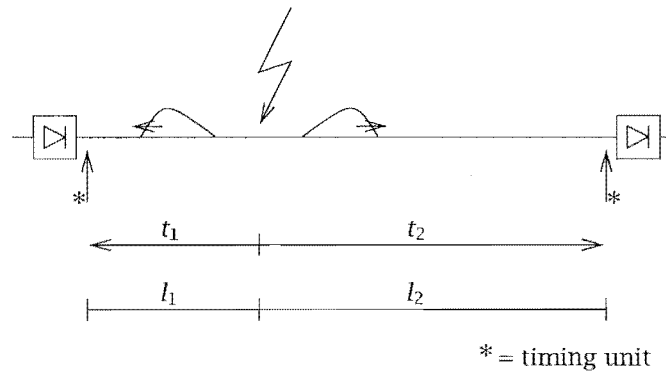


Figure 3.1: Propagation of a line fault on an HVdc system.

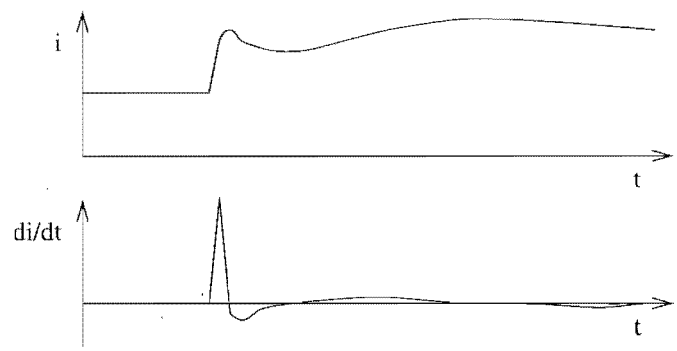


Figure 3.2: Typical waveform of a line fault.

allows customers to mix equipment from different vendors. CORBA (common object request broker architecture[1]) has been proposed as a communication standard for power quality monitoring equipment [27].

Standards have now been developed for methods of measurement and calculation of power quality phenomena, for both the supply side and the load side [59–62].

3.4 Locating Faults on Transmission Lines

Faults on power transmission lines are caused by lightning, equipment failure like insulator breakdowns, or objects or birds shorting out lines. In each case a surge wavefront propagates from the fault location along the line in both directions, as shown in figure 3.1. If repairs have to be undertaken, it can be economically critical to locate the fault position as quickly as possible.

Two different methods used for locating faults in power transmission lines are briefly described below. CHART can be used to implement either of them.

The reflectometry method for locating line faults times the arrival of subsequent surge wavefronts [6–8]. Waves are reflected at the line ends, and the travel time between subsequent surges is indicative of the position where the first surge originated. With each reflection the waveforms become more blurred which makes them difficult to time.

The surge wavefront arrival method for locating line faults was described in [39]. The times of arrival of surge wavefronts at each end of a power transmission line are measured, propagation times are indicative for the fault position. Subsequent reflections are discarded. If the arrival times of two wavefronts at the respective end of the line are equal, the fault position is half way

[1] `http://www.omg.org/`

between the ends. More details can be found in reference [101]. This method requires two timing devices but is also more precise.

Measuring the voltages and currents of all phases of a 3-phase systems at both ends can be used to classify the fault and to calculate the fault position [98]. The digital fault recorder described operates on a 21.5 km long 161 kV power system and uses a sampling rate of 12 kHz. The fault location is derived from an approximation of $\frac{di}{dt}$ from the last samples. The classification indicators are very selective and do not differ significantly with location or incident angle. The location accuracy is not affected by fault resistance and is not much influenced by incident angle.

To avoid the errors introduced by the limited accuracy of CTs a new fault location method was developed [21] which only uses voltage measurements at both ends of the line.

3.5 Accuracy Requirements

This section summarises the accuracies required for a number of different applications. Phadke et al [175] propose the following accuracies:

- State estimation: tenths of a degree (lower accuracy in dynamic estimates could be acceptable).
- Stability monitoring and control: 0.1° is more than sufficient and 1° would be adequate in most situations.
- Fault location: 0.1° when using phasor measurements, 0.5–1 µs when using travelling wave methods for location within 300 m. (This also requires transducers with higher bandwidth.)
- Adaptive relaying: approximately 0.1°.

IEC 61850 defines five accuracy classes for time stamping, which can be used for system specification [63, 64]:

T1	±1 ms
T2	±100 µs
T3	±25 µs
T4	±4 µs
T5	±1 µs

The degree values given are at power system fundamental frequency, therefore 0.1° corresponds to approximately 5 µs. This means that for every of the above tasks except fault location via travelling waves an accuracy class of T4 is sufficient. However, IEC accuracy class T5 allows for example a 1° phase resolution at the 50th harmonic, as discussed in section 4.4 and elsewhere.

Analog filters have components whose values vary with temperature and age. Digital filters do not, therefore their phase shifts remain constant and can be compensated for. This results in greater accuracy.

The accuracy of current measurements can be negatively affected by having to take into account varying load conditions, and by the rated load being smaller than the load under fault conditions. CT shunts typically produce maximum voltage at 20 times the rated load current, that is usually 5 A at full load and 100 A during faults. Normal conditions can therefore produce only a few millivolts [186].

Chen [26] equates harmonic magnitude measurement errors directly with the quantisation interval of the ADC. In fact, the relationship between signal quantisation error and spectral magnitude error is more complicated than this. If quantisation error is the limiting factor, it may be possible to obtain sufficiently accurate measurements with fewer ADC bits than suggested. This issue

is discussed in section 10.2, along with other relevant aspects of the theory behind harmonic magnitude measurement errors.

Sachenko et al [192] point out that the error introduced by the sensor can be several times higher than the error introduced by the system. The sensor error is never included in the error specification by the manufacturer. Sachenko et al propose that the basic features of an intelligent data acquisition system are accuracy, reliability and adaptability, and to compensate for sensor characteristics using neural networks [191, 192]. The particular method of compensation proposed may not always be applicable to power quality monitoring, but the principal problem and possible solution does apply.

chapter 4

System Requirements for Power Quality Instrumentation

When faced with the task of investigating a certain aspect of an electricity supply network, suitable instrumentation must be obtained. The requirements for a particular power quality monitoring system depend to a large extent on the application and the specific answers which are sought in the respective situation. This chapter discusses the significant parameters for such systems, typical values for different types of application, and other relevant issues. A paper has been compiled from this material, which is reproduced in appendix A.7 on page 176. A detailed ordered checklist of the points discussed can be found in chapter 5. This can be used as a template for generating a specific requirement specification for a power quality monitoring instrument.

Systems for power quality monitoring can be broadly divided into three areas of application, each with different complexity:

1. Equipment to measure a single 3-phase system at a single site.

   Such equipment consists of typically 3–8 channels. Time stamping of samples is relatively straightforward because only a single instrument is involved and high time stamping accuracy is not necessary.

2. Equipment to measure multiple 3-phase systems at a single site, such as a substation.

   Such equipment can consist of tens of input channels, and sampling of all channels must be synchronised if a high degree of coherency is required (e.g. when comparing harmonic phases, or phasors). High coherency implies that all channels share a common sampling clock. If channels are not located within the same instrument enclosure, the sampling clock needs to be distributed. A high time stamping accuracy is not necessary.

3. Equipment to measure at more than one site, with each site being in one of the two configurations above.

   The sampling clocks of all involved instruments must be synchronised to a high degree if sample data is to be comparable. In practice this means synchronisation to a highly accurate common time source. This same method of synchronisation is also a possible solution for implementing coherent sampling in the second case above when multiple enclosures are involved.

For each of the above categories, the duration of measurements can vary widely. Examples of short-term applications include confirming a specific operational status, and using a mobile instrument to investigate a specific problem that has started to occur. They are typically characterised

by event-driven recording. An example of a permanent application is continuously monitoring one or more characteristics of the electricity supply network for ongoing conformity, using a fixed installation of instruments.

The characteristics which can be examined depend on both the hardware and the software of the monitor. The hardware defines a permanent maximum limit for the rate at which sample data can be acquired, whereas the software defines the type and method of analysis that is performed on the data. By modifying the software, the monitoring system can be adapted to new and/or specific tasks and new standards for calculating power quality. Generous dimensioning of the hardware allows for a higher level of flexibility with respect to future analysis requirements.

Implications of the above three configurations and other factors present in power quality applications are discussed in the next section. A particular issue is the storage and analysis of prodigious quantities of data, and the importance of reducing this to relevant subsets. Successive sections examine major system parts, which are the input or sampling stage with sample clock generation and time stamping, the software, and aspects of designing custom hardware. No implementation examples are given in this chapter, these are presented for an existing system in chapter 6 and in chapter 11.

4.1 System Aspects

The requirements with the largest impact on overall system design are the need for highly accurate time stamping, continuous operation, and noise issues.

For highly accurate time stamping, an off-the-shelf system is not readily available. Commercial data acquisition systems are examined in section 11.8. It may be possible to build a system from off-the-shelf components. The details of sampling and time stamping, which would have to be addressed, are given in sections 4.4 and 4.5; design choices made there have significant ramifications for the broad system topology.

The use of buffers is difficult because the temporal relationship between sampling clock and samples might be lost. The requirement of continuous operation precludes the use of buffers to extend the maximum sampling time. Data volumes can become high in a short time[1]. Data must be processed immediately to allow for continuous operation. It is essential that this processing causes a data reduction of several orders of magnitude, resulting in a volume of data which can realistically be stored. Data reductions can be achieved by identifying regions of interest, by transforming the time domain data into the frequency domain, and by not storing the FFT result for every mains cycle. Keeping one FFT for each mains cycle is unlikely to be of interest in normal circumstances.

The location of ADCs with respect to the signal source is critical in an electrically noisy environment such as a switch yard or an industrial plant. Shielded cabling is not necessarily adequate, and the electrical characteristics of the sensor, the cable, and the ADC input have to be carefully matched. This can be difficult in practice. Where accuracy is most important, a solution is to locate the ADCs physically close enough to the sensors to prevent significant noise contamination. This generally means that ADCs have to be located in separate enclosures if the signal sources are further apart than a few metres. Multiple enclosures have major implications for coherent sampling (see section 4.4).

Other factors that must be considered for overall system design are the number of input channels, the bus system used, and its maximum card size. The maximum card size determines how many ADCs can be integrated on one card. This has cost implications because in any system design there is a fixed cost associated with each card. The bus system determines the maximum number of cards. The overall maximum number of channels determines the kind of measurement that can be undertaken before a sudden increase in cost, necessitated by a second system, takes place. However, if a limited number of signal sources are in close enough proximity, the maximum number of channels may be of lesser importance.

[1] For example, 40 channels at 50 kHz/s and 12–16 bit per sample result in 330 GB per 24 hours. 1 G = 1024^2

Power supply issues can have an influence on the overall design. For fixed installations, the instrument should be supplied with mains power. For portability and temporary setups, batteries are a good option, although their overall cost (weight, volume, charging, casing) and inconvenience can be significant. Battery life must be sufficient for the required measurement duration. The use of batteries may be the only option when floating the whole system at high potential or for avoiding ground loop issues. Mains supply conditioning needs to bridge supply gaps and filter spikes and surges sufficiently to guarantee reliable system operation. A commercial uninterruptible power supply (UPS) unit could be cost-effective.

The data acquisition units should be networked with another computer for user interface, control, and data storage / backup. It does not make sense to design even a simple system without networking capabilities. A data link allows remote instruments to be controlled, and data to be retrieved from them. It is possible to make the network an integral part of the instrument setup, by sending all sample data to a fast, central computer for processing. If the system is designed for central data processing, sufficient network bandwidth must be available. If basic filtering is performed on the ADC units and final processing on a central computer, network bandwidth requirements can drop to less than 20 kbyte/s/channel.

This central computer could also perform time keeping, with the time being transferred to all ADC systems for time stamping, removing the need for individual external time sources at each ADC unit. However, highly accurate time transfers can not currently be achieved over Ethernet, for details refer to section 11.5.

For mains-synchronised sampling (section 4.4), the mains frequency needs to be determined, e.g. by determining the cycle time of a mains-derived signal connected to a suitable interrupt with a low-pass filter. This can happen on all ADC units individually, or on the central computer followed by transfer over the network to the ADC units. It may be easier to obtain the mains-derived signal at the central computer, depending on the power supply situation of the ADC units. However, doing this centrally introduces measurement uncertainties which need to be considered.

If all processing is to be done by the instrument, then clearly the CPU must be powerful enough to deal with all the design requirements. However, it is expedient to oversize the CPU to allow for future modifications.

Depending on the particular application or intended range of uses, digital inputs and outputs can be considered. If the digital inputs are time stamped with the same mechanism as the analog inputs, event capture channels are available for interfacing to other equipment. Digital outputs can be used for control and signalling functions.

Some applications will require that some or all inputs and outputs are galvanically isolated, e.g. if the whole instrument is floated. Galvanic isolation is also advantageous for avoiding ground loop problems, which would be a safer design for a more general-purpose instrumentation system. Isolating digital I/O lines is relatively straightforward. The components for isolating analog inputs introduce their own frequency and phase response, and their cost to the overall system is likely to be prohibitive. Similarly, overvoltage protection for inputs to e.g. 250 V_{ac} makes a system more robust against accidental misuse, but the costs have to be considered.

As mentioned previously, a commercial system which fulfils all the hardware requirements to the highest level is unlikely to be available. Analysis software tailored to typical applications is not currently available either, irrespective of low or high end systems (software aspects are discussed in section 4.6). A system should be configured from commercially available components wherever possible to reduce cost and development time, especially for a low number of systems. As a general rule, the higher the requirements are, the more hardware will have to be specifically designed for the application. Tradeoffs should be considered. For example, if an ADC card is commercially available for all but the desired input voltage range, it would be more economic to use this card with a pre-scaler in a small external case than to design a new ADC card.

Tradeoffs exist between hardware and software implementation. Larger oversampling factors or transform lengths increase resolution and the computational cost, a wider ADC increases resolution and the hardware cost. The frequency of the system under investigation can be determined

by computation with the measured values, or by using zero-crossing circuitry in connection with a counter.

A number of environmental factors need also be considered. Operation outdoors, unless short term in fine weather, requires adequate protection, whereas standard desktop PC cases are sufficient for indoors. Environmental protection is categorised into classes [40, 58], and class IP64 is needed for outdoors all-weather conditions. Rugged casing makes transport easier, but is unnecessary for fixed installations. Adequate electromagnetic shielding is a design requirement for any instrumentation system. In extreme cases, a minimum distance to strong magnetic field sources may have to be kept.

Relevant government and industry regulations covering aspects such as electrical safety or data exchange need to be taken into account. Electrical safety standards tend to vary between countries, and certification for the target market is frequently necessary. Some countries may accept certification to one of the key standards which are well-known internationally. Numerous industry standards exist for the behaviour of instruments, for example in the area of substation communications [63,64,72], specification of phasor values [71], or the method of measuring and calculating power quality values [61].

A system which compensates for sensor and converter characteristics when reporting values is easier and faster to use. This may require automatic or semi-automatic calibration facilities.

The required level of instrument reliability, which depends on the intended application, affects the overall system design. This should be considered right from the start of the design.

Other features which affect potential deployment of the system include the overall weight and volume, and whether it can be rack or wall mounted.

4.2 Analog-to-Digital Converters

Analog-to-digital converters (ADCs) are built with a number of different underlying principles, which have a large effect on their main characteristics. An overview of the different types is given in this section, followed by a discussion of their suitability for time-critical data acquisition. The different types are:

Parallel, flash An analog comparator is used for each input voltage step, and the comparator outputs are translated into a binary representation with a digital logic network.

This simple design is very fast because of the low number of steps involved for each conversion process. Because the amount of silicon needed rises exponentially with the number of bits, this type is not available in large widths.

Pipelined Consecutive stages of parallel converters increase the resolution, but also add conversion time and lag with each stage. This pipelining can also be accomplished with other ADC types.

Dual-slope The input signal is integrated by charging a capacitor for a fixed time, then the duration of a fixed-rate discharge is measured. Noise is reduced because of the integration, but the conversion process is slower.

Charge balancing A capacitor is charged by the input signal, and discharged with fixed-width pulses and a constant voltage. This effectively is a voltage-to-frequency converter. The pulses per period are then counted.

Successive approximation Starting with the most significant bit, a 1 bit conversion is repeated in a process of binary division, successively arriving at the final result.

Because of the serialised architecture, this type is suited for low and medium sample frequencies. Conversion can be initiated at any time, and must be completed before the next conversion can be started.

Sigma-delta A sigma-delta modulator converts the input signal at high speed into a 1 bit represenation which has an average level equivalent to the input signal. This bitstream is then translated into a binary value with a digital filter.

These ADCs are popular because of their high precision and high resolution which can be achieved with the digital filter by increasing the bitstream frequency. Because of the high sampling rate and cut-off frequency, the analog anti-aliasing filter can be constructed simply. A sample & hold circuit is not needed because the input signal is continuously tracked. The disadvantages of this type are a high demand on the digital filter, and the significant lag until the binary data appears at the output of the digital filter.

The dual-slope and charge balancing integrating type converters have largely been replaced by the sigma-delta type. Very broadly, the three most common types are:

- Parallel: high frequency, low resolution, no lags
- Successive approximation: medium frequency, multi-channel, no lags
- Sigma-delta: low frequency, high resolution, low noise, significant lags

Some of the successive approximation types have a sample & hold (S&H, or track & hold) circuit already integrated on the chip; those without need suitable amplifiers. Parallel converters typically have a S&H circuit included. Sigma-delta converters only need circuitry suitable to drive their input impedance, S&H circuits are not needed because of their design.

Resolution and sampling rate are important factors for any design. The most important factor for a time-stamping data acquisition system is the ability to establish the precise time at which the analog signal was sampled, in relation to a clock pulse which initiates the sampling process or read-out of the value. Further factors which need to be considered for any data acquisition design are the precision and linearity, and for a harmonic analyser, the harmonic distortion introduced by the ADC. As mentioned in section 4.3, higher resolution ADCs require a correspondingly more robust analog design to keep the signal noise below the ADC resolution.

The parallel converters are always suitable for time critical conversions because of their high speed, and are available up to about 14 bit width. Their conversion speed is sub-microsecond, often less than 100 ns, and they are for example used in capturing video signals. The conversion process is started with an external clock signal. The differential non-linearity is < 1 LSB, but the integral non-linearity can be 0.5–2.5 LSB. This type is always suitable for applications with critical timing, but the speed is likely to be much more than what is needed.

Successive approximation converters typically have a conversion speed of 1–10 µs, a differential non-linearity of 0.5–2 LSB, and an integral non-linearity of 0.5–2 LSB, sometimes up to 3 LSB or 6 LSB. Resolution is 8–18 bits. The conversion is started by an internal or external clock signal. Models with more than one input use an input multi-plexer; they tend to have significantly larger non-linearities and lower resolutions. This type is also suitable for applications needing precise time-stamping, and is likely to be the best choice.

Sigma-delta converters are available with resolutions of 10–24 bit after the digital filter. The conversion itself happens with 1 bit. The differential non-linearity is typically 0.05–15 ppm, the integral non-linearity 1–100 ppm. The word rate at the output is very approximately 100–10 000 per second. This type is ideal for industrial applications or multi-meters, where it is important to have high precision and high noise immunity and where a display update rate of 10 per second is sufficient. From the point of view of precise time stamping, the problem with this ADC type is the significant and possibly unknown delays through the digital filters. The conversion time can be much longer than the inverse of the output rate.

New sigma-delta designs exist which incorporate elements of parallelism. For example, the Analog Devices AD9260 is specified with a 2.5 MHz throughput and a 400 ns conversion time. Such short conversion times warrant consideration.

While multi-plexing inputs does not affect the throughput of parallel and successive approximation converters, this is not the case for sigma-delta types. Multi-plexing with these makes a duration equal to the filter delay unusable after each switch while the filter settles.

Using multi-plexed inputs breaks the channel coherency of the instrument because inputs can not be sampled simultaneously. However, this may not be a problem if high-speed ADCs are used and all channels can be sampled within a time shorter than the overall instrument accuracy, or if channel coherency can be effectively re-established by interpolating the data to create time-stamps between samples.

4.3 Input Stage

The input stage of general-purpose instruments needs to accommodate a wide range of source signal types. This is less important for more specific or fixed installations.

The design of the analog front-end and the ADC stage itself is critical with respect to the level of environmental noise resistance. Any design faults here reduce the effective ADC width when the lowest order bits of the ADC fall below the noise floor. This is also especially significant for the connection from the signal source to the ADC input. Appropriate shielding is necessary, especially in a noisy switch yard environment.

The range of input voltage of any instrumentation system is generally fixed, and tends to be in the order of $\pm 1\,V$ to $\pm 10\,V$ for commercially available data acquisition systems.

In power systems, voltage transformers (VT) typically are designed to output 50 V or 110 V at nominal system voltage. Current transformers (CT) are typically designed to output 0.5 A, 1 A, or 5 A at nominal maximum system current. Their output voltage is therefore also dependent on the load. The total VA burden applied to the CT must not exceed its design rating. Care must be taken to ensure that the CT secondary circuit is never open-circuited, and similarly a VT's secondary circuit is never short-circuited. It is expedient to not modify the CT circuit for measurements and to measure the voltage across the existing load instead.

The large voltage range which has to be accommodated has two consequences: external voltage dividers / pre-scalers have to be used with industry-standard data acquisition equipment, and appropriate isolation to mains voltage standard must be provided, in accordance with respective safety regulations.

There is usually no noise problem with VTs because of the high signal level. The possible exception are measurements for fault location, where the fault is close to the measurement point. However, if the measurement location is typically in a substation or switch yard, any nearby fault should not be difficult to find without instrumentation.

Automatic gain adjustment (auto-ranging) can increase the effective dynamic range of the ADC, but has the following disadvantages:

1. The switching is not instantaneous, therefore some period of time is not captured and badly distorted and/or lost. The sample time stamping needs to be able to cope with this. The main factor of the switching delay is the reaction time of the software.

   If transient capture is a part of the system's objectives, switching must be implemented in a way that does not interfere significantly with the transient waveform.

2. If suitable solid state switches can be sourced, the input range needs to be switched in sync with the sampling.

3. The analog circuitry required to implement this introduces further distortions and needs careful shielding to minimise the pickup of environmental noise.

4. Adjustable input ratios are not usually provided with commercially available ADC cards, or at least not with the desirable scale factors. If a power monitoring system can be otherwise implemented completely with commercially available components as outlined in chapter 11, adding a pre-scaler as custom-built hardware adds difficulties to the design.

Commercial systems are specified with respect to their performance for signals connected at their inputs. It can not be assumed that the signal at this point is a correct representation of the quantity to be measured, because the transfer characteristics of the sensor (transducer, current probe, etc.) must be taken into account. If these characteristics are known, they can be compensated for by the data processing software. For the most accurate measurements, the characteristics should therefore be obtained. If saturation is reached, compensation is not possible. The characteristics of CTs and VTs have been optimised for power system frequency. The output of harmonic frequency signals will suffer some degredation in both amplitude and phase. A system which is able to auto-correct for sensor characteristics, or is able to at least establish the sensor characteristics automatically with simple manual interaction, is user-friendly. However, while this may work well for, e.g. current clips, the scope for characterising CTs or VTs already in service is more limited.

The signal at the ADC stage must be limited in frequency to twice the sampling frequency, or the digital representation of the signal will be inaccurate. If this condition is not implicitly met by the characteristics of the source signal, a suitable low-pass filter must be used. The amplitude and phase error of such a filter directly adds to the system's total error. Because this filter's response is well-known, its effect can be compensated for in harmonic analysis.

4.4 Sampling and Sample Clock Generation

Two of the key parameters of any data acquisition system are the sampling rate (frequency) and the effective size of each sample when digitised (bits). The minimum sampling rate depends on the highest order harmonic that is considered to be of interest, and the degree of accuracy to which the harmonic phase information is desired. The relationship between sampling rate, ADC width, and obtainable harmonic phase is examined in chapter 10.

To recover harmonic magnitudes, the minimum sampling rate is twice the frequency of the highest-order harmonic to be recovered, e.g. 5 kHz for the 50th harmonic of a 50 Hz system. In practice, choosing a slightly higher value would avoid any marginal error. For 60 Hz electricity supply systems, frequencies need to be multiplied by 1.2. Sampling rates in this range are sufficient to calculate total harmonic distortions (THD).

With oversampling, the design of any analog low-pass anti-aliasing filter for the ADC input is simplified. The main advantage is that oversampling, in connection with a digital FIR (finite impulse response) filter, reduces quantisation noise introduced by the ADC. A higher rate of oversampling has the potential to increase the effective ADC resolution (number of bits), but also increases the bandwidth and FIR filter computational needs. 8× oversampling (40 kHz) should not present great difficulties in terms of additional computational requirements for the FIR filter at these comparatively low sampling frequencies.

ADC widths of 12 bits are widely available. Widths of 8 bits are common for very high sampling rates in the megahertz range, these high rates offer no advantage for power quality instrumentation. Widths of 16 bits are also commonly available. Widths in excess of 16 bits are relatively uncommon because noise interference problems become difficult to control, and are not likely to be cost effective. 12 bits provides a good compromise, while 16 bits will satisfy the most stringent requirements (see chapter 10).

The digital signal processing computational requirements are unaffected by the ADC width, because the minimum data type which can be handled is 4 bytes (type "single" in the C programming language), assuming the calculations are performed in floating point format. It would be possible to perform calculations with sufficient accuracy with integer arithmetic only. A 16 bit arithmetic would be sufficient for 8 bit ADCs, however 12 bit ADCs would require a 32 bit arithmetic. The relatively small advantage in execution speed of integer over floating point numbers in modern processors needs to be weighed against the time and effort of having to software engineer all computations (including DFTs) in integer.

A certain amount of control over the sampling clock generation is desirable. Commercial systems always allow programming of the sample rate directly, based on a crystal oscillator. Supplying

the sampling clock externally is also a standard feature. The latter is necessitated by the need to sample at identical frequencies across a number of channels, as well as by sampling with a known phase relationship across channels. This becomes an issue as soon as channels are spread over more than one data acquisition printed circuit card.

For some applications, it is important that the sampling clock skew between channels is kept to a minimum. This includes applications for harmonic state estimation or harmonic power flow, where accurate harmonic phase information must be obtained. It is also important for transient analysis. Using one of the ADC cards to generate the sampling clock, and supplying this as external signal to the other cards, is a standard feature of commercial ADC cards and solves this part of the clock skew problem. Clock skew is also created by multi-channel ADC cards which are multiplexed. These cards sample all channels sequentially. The clock skew introduced by multiplexing may however be deemed negligible if the total time taken to sample all channels is much smaller than the sampling period. If this is not the case, one ADC per channel should be opted for.

Certain types of ADC require a sample-and-hold (S&H) circuit before the ADC input which stores the present voltage level for the duration of the conversion. This prevents a corruption of the digitised value with fast-changing signals. Manufacturing imperfections in this circuit introduce errors into the measurement. Because S&H-circuits are generally integrated into the ADC converter chip, their error contribution is specified as part of the ADC's error specification.

Of interest for electricity supply applications, which usually involve the computation of Fourier transforms, is the feature to specify a sampling rate which is a multiple of the fundamental frequency. This eliminates rectangular windowing issues with the FFT calculation, and simplifies software design and processing needs. Sampling clock generation is therefore characterised by:

1. Samples per second
2. Sampling per an externally supplied clock signal
3. Samples per mains cycle

Sampling at a multiple of the fundamental frequency can be achieved by continually changing the programmed sample rate, if the actual mains frequency is known to the data acquisition system. However, the resolution with which the sampling rate can be programmed will need to be considered. It has to be decided whether it would be sufficient if the sampling rate could e.g. only be set in terms of whole samples per second.

Flexibility in when the system is operational or stopped is discussed with other software aspects in section 4.6.

4.5 Time Stamping

Time stamping, or time tagging, of sample data is of much interest. Two cases have to be distinguished: single-site and distributed measurements. If a single instrument with a number of channels is involved, coherency is generally inherent if the input channels are not multiplexed. The resolution of the time stamping is as provided by the system, and the accuracy is largely irrelevant because all samples can be related to each other. The situation is similar for multiple systems at the same site if they share a common sampling clock. One system is the time master for the others. The characteristics of the time transfer from the master system can not be neglected if the delay of the time transfer introduces an unacceptably high clock skew, but accuracy remains irrelevant.

For a distributed measurement setup, all systems need to be synchronised. For the purpose of comparing data between these systems, it is sufficient to synchronise all involved systems with each other. This is internal synchronisation. A relationship of system time to "outside" time is only needed for comparison with data obtained elsewhere, if this is of interest. However, a

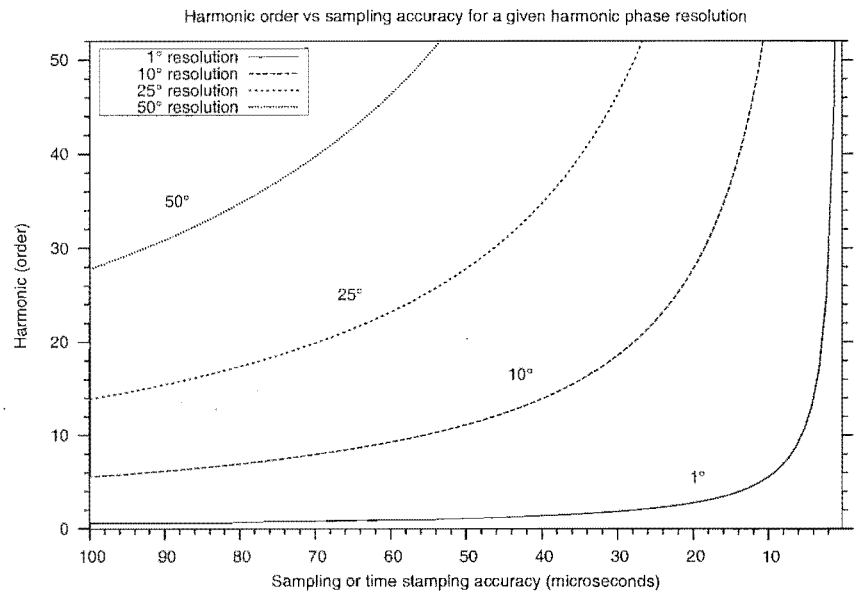


Figure 4.1: The maximum harmonic order which can be resolved with a given timing accuracy and phase angle resolution. This graph is based on a 50 Hz system. For a 60 Hz system, divide the harmonic order by 1.2. The same graph plotted on a logarithmic scale is given in figure 5.1.

time transfer from a master system to all the other systems is not practical or economic. The only practical solution is to synchronise all systems to an external time source. This is external synchronisation. The accuracy of this synchronisation to an external source, and therefore the accuracy of the time stamping, directly limits the resolution to which sample data from different measurement locations can be compared. If different data acquisition systems use different external time sources, the error between the external sources has a direct influence. For example, in case of the GPS, synchronisation is better if receivers are tracking the same satellite.

The accuracy of the sample time stamping translates into a limit for the degree to which harmonic phases can be compared. Figure 4.1 relates the harmonic order and the time which correlates to a given fraction (angle) of that harmonic. For example, 25° of the 14th harmonic correlates to 100 µs, whereas 1° of the 50th harmonic correlates to approximately 1 µs. The only practical and economic way to achieve a timing accuracy to this level is by utilising satellite time receivers.

Flexibility is increased if time stamping remains functional in the absence of a signal from the external time source, although a reduced accuracy should be permissible. In practice, the additional error may increase with the duration of the time reference having been unavailable.

Time stamping of external events should be performed with the same accuracy as the time stamping of sample data. The standard IEC 61850 [64] defines 5 different accuracy classes from 1 ms to 1 µs (see sections 5.3.4 and 3.5).

4.6 Software

Individual analysis of particular sample data can be performed with general-purpose mathematical software like Matlab (a large range of such tools exists, e.g. Octave[2] or Scilab[3]), but this is not suited for permanent unsupervised operation, or generating any kind of alerts. A higher level of automation is desirable. Possible functional requirements for the software therefore are:

[2] http://www.octave.org/
[3] http://www.scilab.org/

1. Control of the beginning and end of sampling, both manual and automatic. A list of times when sampling starts and stops should be configurable, and this should include the possibility of repetition (e.g. every N seconds, hours, days or weeks). Activation or deactivation of sampling by an external signal may be required, especially for interaction with other equipment.

2. Methods of sample data processing. The instrument should be pre-loaded with a range of commonly-used filter and power quality analysis functions, including e.g. total harmonic distortion, flicker, dips and sags.

3. The ability to set thresholds and generate trigger conditions. Power quality parameters should be calculated according to applicable industry standards, and cause a trigger condition when preset thresholds are exceeded.

   This may require additional hardware or software interfaces to other equipment.

4. The ability to complement the existing built-in data processing functions by user-supplied functions, to allow for tailoring to specific conditions and problems. The user should only be concerned with programming the function itself, all other programming aspects should be of no concern.

5. The ability to configure storage of processed data for the short and long term, and to specify which data should be stored at all. It should be possible to specify a maximum storage size per result type, so that when this is reached, the oldest data is deleted first. Data which has not been downloaded by then and archived, whether this is automated or not, would be lost permanently.

   This should be organised in levels, comprising different time spans and levels of detail.

6. Compensation for sensor characteristics. For accurate measurements, it is necessary to compensate for the characteristics of transducers and probes. If this is desired, the system software needs to be able to deal with transfer characteristics supplied by the user in some suitable format.

7. Storage of time-domain data. If the ability to re-process the sample data at a later time is required, the sample data needs to be stored. Simplification of post-processing and reduction of data volume can be achieved by storing the data after it has passed noise and anti-aliasing filtering and sensor charateristic compensation (item 6). To limit resource requirements, a maximum duration for which time-domain data can be stored by the system per a certain interval should be specified.

8. Storage of data for a configurable time preceeding and following a trigger condition. Storage can be arranged in a similar way to that described in item 5. Whether the storage of time-domain data (item 7) is also needed has to be decided. As the storage of this data as a result of a trigger condition could mean an increased importance for this data, storage of this data in a separate area may be advantageous.

   This particular function is important and has large documentary and evidential value in case of serious faults like blackouts, when the cause needs to be established. Information about the state of the distribution network before the event is essential.

9. Remote control all the functions listed here via data network. TCP/IP is universally used for this and therefore guarantees interoperability with LANs.

10. Data reduction, as a general principle. It is impossible for anyone to work through the massive amounts of data which can be generated. Software must be capable of allowing the user to specify which information is important and which should be discarded.

11. Determination of mains frequency for controlling the sampling, replacing any equivalent hardware solution.

12. Encryption for control and data communication. This is important over shared networks to prevent unauthorised control of the instrument.

13. Well-designed user interfaces, to aid efficient work flow and reduce accidental misconfigurations. This is also true for control elements of the hardware.

14. Scriptability of the controls and data storage allows a high-level programmable access to the instrument and an integration into multi-instrument configurations. It also allows automated configuring.

Apart from functional requirements of the software, general aspects of development, implementation and support need to be considered. Multi-tasking capabilities aid in modularising the software and are a standard feature.

The choice of operating system (OS) has long-term implications. Aspects which need to be considered include long-term availability and support, and to what extent real-time features are required. Availability should not become a problem with any supplier with a large market share, and is implicit with open source software. Support by the supplier is not needed if support from third parties is available as well or instead. The ability to hire skilled staff should be considered. The more specialised the operating system, the more difficult this becomes.

If a real-time operating system for embedded systems is used, specialised tools are needed for development, and the total development time is likely to be longer, resulting in a higher cost. Average throughput can be relatively low. Using a general-purpose operating system results in a large number of low-cost development tools, and the software can for the most part be tested on any PC. A large amount of software is readily available and may be incorporated into the design (license issues permitting). A middle way between real-time and general purpose is a general purpose operating systems with a real-time core.

The portability of software to different platforms is an advantage in heterogenous environments, especially when hosts are networked. When designing data storage formats, the compatibility of binary data with big-endian and little-endian processors is key to successful data interchange. Data exchange in text format such as XML has become popular for this reason, but the associated overheads (increased size and conversion time) have to be allowed for.

For the archiving of processed data a data base can be used, allowing for convenient access to past information. This adds a piece of complex software to the system, and another cost factor (unless a free data base is used). Equivalent functionality can probably be obtained by organising the storage of files in a specified way.

An instrument of this complexity calls for a large amount of user documentation. This also applies to any custom-made hardware, and the system as a whole.

4.7 Designing Custom Hardware

Designing custom hardware should be avoided whenever possible, but becomes necessary when standard hardware is not available. The decision involves making tradeoffs between development time, cost and functionality.

It is more economic to design custom hardware to connect to an established interface or bus system. The interface should be chosen carefully because it will have long-term implications for future developments. Open bus systems which are supported by more than one manufacturer are preferable, this is true for both the hardware and the software running on that hardware.

Ideally, a platform for development of custom hardware would fulfil the following requirements:

1. be supported by multiple vendors

2. have sensibly-controlled specifications

3. have a sufficient throughput for processing, storage, and downloading of data

4. have multi-master capabilities
5. offer a minimum card size for user hardware
6. offer a sufficient number of slots
7. allow a means to carry user-defined signals between user-designed cards
8. be maintainable (also true for software)

Any method of transferring time from one card to another through hardware would require some sort of bus system. It is possible to wire ribbon cables along the front panels, or through the rack on the inside after solving the problem of inserting the cards with the cable attached, but this is rather messy. A convenient and tidy way of interconnecting cards is to use backplanes. For this kind of purpose, the Multibus II provides an abundance of free lines. Free lines are also provided by VME and PXI. For a compact PCI system, 64 bit versions exist; using only 32 bit cards, the remaining lines on the backplane can be otherwise occupied (with suitable connector keying and/or pinout choice to prevent accidental damage).

If time stamping is implemented with the use of hardware counters, the full width of the counter can be transferred via a bus to the ADC cards. Alternatively, if only a few bus lines are available, the counter can be duplicated on the ADC cards, requiring only a clock and a reset line to be transferred by some means.

It is useful for development, production testing, and in the field if some basic means are provided which can quickly and easily verify the functionality of the hardware.

chapter 5

System Requirements Specification Template

System requirements for power quality monitoring have been discussed in detail in chapter 4. This chapter comprehensively summarises the points that are raised there, for use as a convenient checklist. It effectively provides a template for the system requirements of a power quality instrumentation system, and is intended to assist those who are drawing up a specification for such a system.

The checklist begins with an overview of the purpose and use of such a system, followed by a list of issues which affect overall system design. These issues are followed by requirements for data acquisition, processing and storage. Communications between parts of the system are considered next, and then options for operator interaction with the system. Then environmental considerations are listed, and finally, issues associated with design and maintenance of the proposed system's software and hardware.

5.1 System Overview

A data acquisition system for power quality monitoring typically performs some or all of the following functions:

- Calculation of power quality parameters, e.g. total harmonic distortion
- Determination of harmonic power flows
- Accurate time stamping of data for distributed measurements
- Fault location
- Response to line faults
- Analysis of transients
- Continuous resp. permanent operation without supervision

A data acquisition system capable of performing the above functions consists of the following stages:

$$sensor \rightarrow ADC \rightarrow processing \rightarrow storage \leftrightarrow analysis$$

The sensor stage comprises any equipment which provides signals to be measured, e.g. CTs and VTs, or current clips or probes. The ADC stage may also include a pre-scaler which adapts the input range of the ADC to the range of the sensor, possibly with auto-ranging capability, and a signal conditioning filter. The ADC itself converts the analog signal into a digital representation, and for reasons of noise immunity should be located close to the signal sources. The location of the ADC stage is likely to have a large influence on overall system layout.

The processing stage performs further digital filtering, and performs the bulk of the computations on the acquired data. Information that is deemed to be of interest to the user is kept, the rest is discarded, reducing data rates significantly. Data of interest is stored, and can be analysed further as required. This can happen immediately or at a later time.

Time stamping of sample data is performed at the ADC or processing stage, depending on requirements. For the most stringent applications, a time stamping accuracy of 1 µs or better can be implemented. Interfaces to external time sources, e.g. network time servers or GPS receivers, are provided.

For distributed measurements, multiple instruments can be linked together and controlled from a central location.

5.2 System Configuration Considerations

This section provides an overview of the aspects that have the greatest effect on overall system design. Details on parts of the system are given in later sections.

5.2.1 System Complexity

The system configuration can be broadly divided into three classes as described below. The specification should identify the appropriate applicable class.

1. Measuring a single 3-phase system at a single site *(A)*.
   (a) Typically 3–8 input channels.
   (b) Time stamping accuracy requirements are low.
   (c) Precision requirements are relatively low.
   (d) Requirements of coherency between channels depend on application. Harmonic phase comparisons require much higher coherency.
2. Measuring multiple 3-phase systems at a single site *(B)*.
   (a) Tens of input channels.
   (b) Time stamping accuracy requirements are low.
   (c) Precision requirements are relatively low.
   (d) Requirements of coherency between channels depend on application. Harmonic phase comparisons require much higher coherency.
   (e) If possible, all channels in one instrument enclosure; otherwise, common distributed sampling clock for high degree of coherency.
3. Measuring at more than one site, one or more 3-phase systems at each site *(C)*.
   This assumes data from multiple sites is to be compared; if not, cases *(A)* and *(B)* above apply.
   (a) Typically 3–8, or tens, of input channels at each site.
   (b) Time stamping accuracy requirements are high, and must exceed the coherency requirement.

5.2.2 Duration of Measurements

The instrumentation system may be required to carry out measurements in one or more of the following modes of operation, with their specific key characteristics:

1. Short-term
    (a) Portable
    (b) Lightweight
    (c) Versatile, configurable for different applications
    (d) Battery life
    (e) Networking for remote control
2. Permanent
    (a) Fixed installation — rack or wall mountable
    (b) Durable
    (c) Mains power supply
    (d) Battery backup for temporary power failures
    (e) Networking for remote control and data transfer
3. Continuous
    (a) Continuous processing to reduce data quantity
    (b) Sufficient data storage for the quantity following processing
    (c) Data transfer
    (d) Networking for remote control and data transfer
    (e) Versatile, configurable for different applications
    (f) Mains power supply or battery

5.2.3 Electromagnetic Environment Considerations

Minimising impact of electrically noisy environment by placing ADCs within a few metres of signal source is a prime consideration. One or more of the following physical configurations may be relevant:

1. All ADCs in one instrument enclosure if sources close enough.
2. ADCs in multiple enclosures if sources too far apart. Requires:
    (a) Distributed sampling clock
    (b) Networking of ADCs and central computer(s)
3. Connection of ADC units to control/processing unit

5.2.4 Communications

Depending on the system and physical configurations adopted, the communication media and protocols between the constituent units must be appropriately chosen. Analog signal transmission should be carefully screened and kept to a minimum.

1. Shielding of analog lines
2. Electrical isolation of all interconnection cables
   (a) Fibre-optic
   (b) Isolation transformers
3. Networking protocol and medium, and bandwidth for data and control
   (a) Within a single site: Ethernet (100M or 1G)
   (b) Between sites: PPP over modem and telephone lines, ADSL, ISDN, frame relay, etc.

5.2.5 Separation of the Main Data Processing Unit from ADC Stage

Factors which need to be considered when performing the main data processing e.g. in the control room with readily available electricity supply and off-the-shelf computers, while keeping the ADC units close to the signal sources, include:

1. Amount and type of pre-processing on ADC units
   (a) Analog filtering
   (b) Digital filtering
   (c) Compensate for non-linearities introduced by analog or digital filtering
   (d) Compensate for transfer characteristics of the particular sensor / signal source connected to each channel. (This may not be possible with limited amounts of digital filtering by the ADC units.)
2. Time stamping at ADC (more accurate) or central computer
3. Networking bandwidth between ADCs and site-central computer(s)
   (approx. 10–15 kbytes/s/channel are required without oversampling)

5.2.6 Hardware vs. Software Implementations

A number of tradeoffs exist between implementation of functionality in hardware or in software.

1. Higher ADC width increases resolution and dynamic range
2. Larger oversampling factor increases resolution, and simplifies analog low-pass filter design
3. Larger transform length increases resolution
4. Higher sampling rate increases computational requirements
5. Higher sampling rate requires more network bandwidth for data transfer to main processing unit
6. For mains-synchronised sampling, deriving the fundamental frequency from the sample data requires some computations but dispenses with the mains reference signal (and associated interface / isolation etc. costs)

5.2.7 Power Supply

The sources of electrical energy for the instrumentation system will be dependent on the particular application and situation on site, and include the following:

1. Mains supply
    (a) Availability on site
    (b) Galvanic isolation
    (c) Filtered for transients, to a sufficient level for reliable system operation
    (d) UPS, to cover dropouts sufficiently for reliable system operation
2. Batteries
    (a) Backup for temporary mains failure
    (b) May be required for remote locations
    (c) May be required if floating system at high potential
    (d) Must last long enough for longest measurement
    (e) Charging
    (f) Weight and volume
3. Solar panel(s)
    (a) Space requirements
    (b) Power output
    (c) Battery backup

5.2.8 Development Constraints

Trade-offs between development time, cost, achievable functionality, and commercially available functionality need to be made. For example, off-the-shelf ADC card with custom-made pre-scaler vs. custom-made ADC with integrated pre-scaler.

1. Time
2. Budget
3. Quantity of units
4. Off-the-shelf components or custom-designed hardware.
5. Software or hardware solutions for specific aspects
    (a) Hardware gives absolute limit on data acquired
    (b) Software determines available methods of analysis
    (c) Software can increase flexibility
    (d) Software has no per-unit cost, but a high cost of production

5.2.9 Compliance With Relevant Standards

Some standards are mandatory; others are only required if the system is to be sold, or used in certain conditions. Standards vary between countries, giving a comprehensive list here is impractical. Standards in these areas need to be considered:

1. Safety (electrical, other)
2. Electrical interfaces, for interoperability
3. Data exchange and storage, for interoperability
4. Accuracy (measurement, time stamping)
5. Implemented functionality
6. Environmental protection (e.g. IEC 60529 class IP20, or IP64)
7. Electromagnetic interference
8. Electromagnetic compatibility (e.g. IEC 61000 / EN 61000)
9. Casing / enclosures, where applicable

5.2.10 System Calibration

The error introduced by the sensor can be several times larger than the error of the instrument. A system which is able to include the error of the overall measurement process, including sensor, in its reported output is easier and faster to use. This may require additional hardware components:

1. Automatic calibration, excluding sensor(s)
2. Automatic calibration, including sensor(s)
3. Manual sensor calibration
4. Suitable signal generator to use with calibration provided by the instrument

5.2.11 Overall Volume and Weight

Overall system volume and weight affects deployability, and is itself affected by environmental (moisture, temperature, dust) and electrical requirements.

5.2.12 Reliability

The intended application, and the potential amount of damage in case of system failure, dictates the required overall level of system reliability, which is a function of the reliabilities of:

1. Hardware design
2. Components
3. Manufacturing
4. Operating system software
5. Application software

5.2.13 Hardware Resources

Sufficient hardware resources need to be provided for:

1. Data processing and storage as per section 5.4
2. Data download from the instrument such that continuous instrument function is not affected
3. Time stamping and sample clock generation (this is dependent on their implementations)
4. Time domain data handling as per section 5.4.5 item 8

5.3 Data Acquisition Modules

This section lists the issues that must be considered for the data acquisition stage of the analysis system, including the nature of the sample input, analog-to-digital converter design, sampling approaches and time stamping of sample data.

5.3.1 Input Stage

The input stage is the interface to the signal sources, and must match the characteristics of the source. Compensation for the characteristics of particular transducers can be performed with software, refer section 5.4.6.

1. Accommodation of a wide range of signal source types
   (a) Standard transducers
   (b) Specialised voltage and current transducers
2. Input voltage range
   (a) Sources: 100 mV to 1000 V
   (b) Commercial ADC cards: typically 10 V
   (c) Voltage dividers required?
   (d) Voltage dividers integrated into acquisition module, or in their own separate enclosure
3. Auto-ranging, if disadvantages are acceptable (delay before range changes, problematic with transients, adds to noise level, adds design complexity in both hardware and software)
4. Low-pass filter, if necessary
5. Electrical safety: signal lines may have to be isolated
6. Determining frequency of distribution line/feeder/system

5.3.2 Analog-to-Digital Converters

The analog-to-digital converter is the key component of every data acquisition system, and together with oversampling factor, sampling rate, and digital filter and transform characteristics largely determines the corner points of the instrumentation system. (Refer to chapter 10 for the details of the relationships between these factors.)

1. ADC width; typically 8, 12, or 16 bits
   (a) Required width depends on required accuracy of phase recovery, sampling rate and oversampling
   (b) Doubling ADC width halves quantisation noise
2. ADC type
   (a) Successive approximation
   (b) Parallel (faster than needed?)
   (c) Sigma-delta (difficult for time stamping)
3. Sampling rate
   (a) For harmonic analysis, minimum is 2× frequency of highest-order harmonic to be recovered
   (b) For transient analysis, depends on desired temporal resolution
   (c) Quantisation noise reduces with increasing FFT length
   (d) Oversampling factor; reduces quantisation noise

5.3.3 Sampling

To support a wide range of applications, it is desirable to have flexibility in configuration of sampling, including:

1. Sampling control. Desirable choices may include:
   (a) Setting number of samples per second
   (b) Setting number of samples per period of an externally supplied reference signal
   (c) Setting number of samples per mains cycle (to remove windowing issues in FFT)
   (d) Continuous sampling
   (e) Periodic sampling
   (f) Sampling start/stop automatically triggered by events
   (g) Manual start/stop
   (h) Delayed start/stop
   (i) Remote configuration and sampling start/stop
2. Determining mains frequency
   (a) For example by determining the cycle time of a mains-derived signal connected to a suitable interrupt with a low-pass filter
   (b) At each ADC unit or centrally if channels are operated with a high level of galvanic isolation
   (c) By computation from the sample data

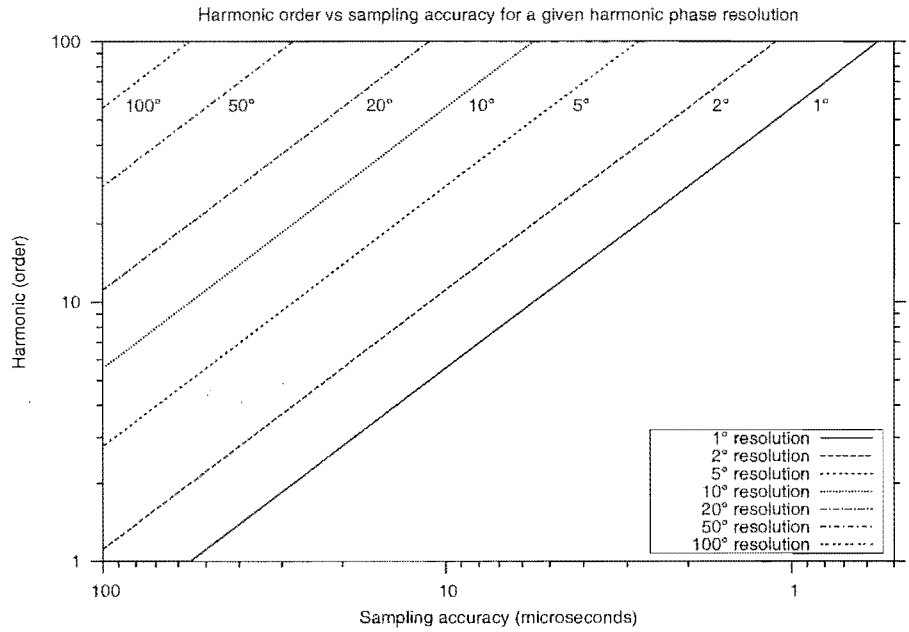


Figure 5.1: The maximum harmonic order which can be resolved with a given timing accuracy and phase angle resolution. This graph is based on a 50 Hz system. For a 60 Hz system, divide the harmonic order by 1.2.

3. Oversampling
   (a) Basic noise filter
   (b) Anti-alias filter
4. Sample clock source
   (a) Master sampling clock on one ADC card (for single instruments)
   (b) External (for multiple instruments and single instruments if channels are on multiple cards)
   (c) For multiple sites, synchronisation to satellite time signals (e.g. GPS)
5. Sampling skew between channels (needs to be minimal for harmonic phase and transient analysis). To consider:
   (a) Multiplexing: only if all channels are sampled in $\ll$ sampling period; otherwise 1 ADC per channel
   (b) For external clocks, symmetric sample clock transfer to each ADC (e.g. identical cable lengths)

5.3.4 Time Stamping

The performance of the time stamping is the key factor for distributed measurements. A suitable means needs to be chosen for supplying a reference time (e.g. UTC) to the system.

1. Time stamping accuracy required for harmonic phase recovery
   (a) For a single instrument, coherency between channels should be implicit in the design. Accuracy is not the relevant factor (unless data is compared with other data not obtained by this instrument).

(b) For multiple instruments at a single or multiple sites, accuracy of time stamping is directly related to the limit of harmonic order and phase angle (refer to figure 5.1).

(c) Accuracy class of time stamping according to IEC 61850 part 5 [64]:
T1 ±1 ms
T2 ±100 µs
T3 ±25 µs
T4 ±4 µs
T5 ±1 µs

2. Time stamping accuracy required for other applications

(a) Fault location accuracy (specified in metres) is related to time stamping accuracy

3. Source of reference time

(a) External time receiver connected to the instrument (e.g. GPS)
Possible connections required: RS-232C, pulse-per-second, power supply

(b) Integrated time receiver
Possible connections required: antenna

(c) External precision time source connected to the instrument (unlikely to be cost-effective)

4. If the reference time source becomes unavailable, the system should remain functional at reduced accuracy for a certain time

5. Capturing and time stamping of events

5.3.5 General-Purpose Digital Inputs and Outputs

Digital inputs and outputs can be used for controlling or interfacing to other equipment, or to provide a simple user interface (system status, etc.) at the instrument itself. These are specified by:

1. Number of inputs and outputs
2. Electrical isolation
3. Maximum clock speed

5.3.6 Shielding / EMC

The shielding of the ADC units against relatively high levels of electromagnetic interference should meet these minimum standards.

1. Electromagnetic noise in the environment must not affect the operation of the digital electronics or degrade analog signals.
2. To make full use of ADC width, noise level must be below quantisation interval of ADC.

5.4 Data Processing and Software

The software requirements of a power quality monitoring system depend on the particular application, but should typically implement the functionality listed in this section.

5.4.1 CPU

Required CPU processing power depends on:

1. Built-in algorithms
2. Support for user-programmed algorithms
3. Reserves left for future extension of algorithms
4. Previously-made choices for hardware vs. software implementation of functionality

5.4.2 Built-in Functions

The flexibility of a monitoring system increases with the number of analytical functions which are implemented. Common functions include:

1. Anti-alias (low-pass) filter, down-sampling filter
2. FFT (frequency analysis)
3. THD, flicker, dip, sag and other common power quality parameters. These should be implemented according to applicable standards.
4. Transient detection
5. Comparison of a number of values with preset thresholds

5.4.3 User-Definable Functions

Allowing users to program specialised algorithms increases system flexibility. A suitable mechanism to interface the user-supplied part to the system needs to be provided and documented.

5.4.4 Trigger Conditions

Thresholds applied to the result of the data processing described above can generate a trigger condition or event, causing a certain system action. Possible actions include:

1. Storage of data before and after the event
2. Activation of audible or visual indicators, or pop-up windows on a computer screen
3. Archival logging of parameters and data associated with this event
4. Notification of operators via electronic mail
5. Generating control commands for other power systems equipment

5.4.5 Data Storage

There are a number of issues associated with storage of the sample and/or processed data. A reduction of the data rate for storage is as important as eventual automatic removal of "old" data for continuous operation.

1. Storing processed data selectively
2. Required storage volume
3. Storage rate
4. Storage media: hard disk, flash memory, etc.
5. Marking of selected data as "important" for long-term storage
6. Automatic removal of oldest data, unless marked for permanent safe-keeping
7. Storage of data from a configurable time before and after a trigger condition
8. Time-domain data e.g. for transients (much higher rate); up to maximum volume, or up to maximum duration followed by minimum pause
9. Long-term archival storage

5.4.6 Compensation for Sensor Characteristics

Obtaining accurate measurement results requires taking into account any sensor non-linearities. For ease of use, sample data can be corrected before display and storage. The instrument's software should support:

1. List or data base of sensor transfer functions which can be called upon
2. Implementation of more elaborate algorithms for compensation of sensor characteristics
3. Transfer characteristics that can be linked to any channel
4. Automatic or semi-automatic calibration of sensor(s) and system

5.5 Communications

The level of networking required depends to a large extent on the basic system configuration choice made in section 5.2.1, and the desired level of real-time interaction for distributed operation. This section lists the functions which the data network should be able to perform.

5.5.1 Remote Management

The ability to control the instrumentation system remotely via network is a matter of convenience. For large values of "remote", it becomes a necessity.

1. From control room, for safety reasons, to avoid entering switch yards unnecessarily
2. From off-site
3. Manual control of sampling
4. Troubleshooting of system

5.5.2 Interaction With Other Monitoring and Control Equipment

If the system is to be integrated into a site-wide monitor and control strategy, exchange of information between systems from different manufacturers needs to be ensured.

1. Compliance with standards for power systems information exchange, e.g. IEC 61850 [63, 64]

5.5.3 Control by External Signal

In connection with other control systems, triggering data recording or data storage by an external signal can be useful for investigating irregular phenomena.

1. Automatic triggering of sampling
2. Automatic triggering of storage

5.5.4 Encryption

Encryption is mandatory when connections to the instrumentation system go over a public network, i.e. the internet, to prevent unauthorised control of the system. The relevant factors are:

1. Strength of the encryption
2. Encryption of control streams (mandatory)
3. Encryption of data streams (can be omitted if the recorded data is not considered to be of a sensitive nature)
4. Additional computational load

5.6 User Interface

The user interface determines how the operator interacts with the measuring instrument, and is a major aspect of the instrument's usability. It should be as user-friendly as possible. With careful selection of default settings, user configuration can be minimised.

5.6.1 Instrument Controls

The methods through which operators will interact with the system need to be determined. Possible options, which are not mutually excusive, include:

1. Function-specific buttons
2. Menu structures
3. Command-line interface on PC
4. Graphical interface on PC
5. Control of entire instrument via scripting

5.6.2 Instrument Setup

User-friendliness of instrument controls is important more so for general-purpose portable instruments than those permanently installed.

1. Channel configuration can be applied to multiple channels
2. Current setup status can be viewed easily in its entirety

5.6.3 Display

Measurement results and the operational status of the instrument need to be presented to the operator. Variables to display and the means of displaying them include:

1. Currently applied settings
2. Current and past events
3. Current and past status
   (a) LEDs
   (b) Logging
4. Currently recorded time + frequency domain values
5. Troubleshooting information
   (a) LEDs
   (b) Logging
   (c) Data being currently collected
6. Results in graphical form
7. Means of display
   (a) LEDs on the instrument
   (b) LCD on the instrument
   (c) Computer monitor (cathode ray or LCD)

5.6.4 Documentation

An instrument of this complexity requires a significant amount of documentation. The instrument itself is a mixture between an oscilloscope-type instrument and a computer, which should be taken into account by the presentation used for the documentation.

1. User manual
2. Quick reference
3. Online help
4. Statutory requirements, especially safety warnings

5.7 Environmental Considerations

The level and type of protection against environmental factors determines in which situations the instrument can be deployed.

5.7.1 Physical Environment

The basic types of physical protection to consider are:

1. Weather-proofing, e.g. IP64
2. Ruggedisation
3. Operating and storage temperature ranges

5.7.2 Electromagnetic Environment

Electromagnetic fields in an industrial environment can adversely affect electronic devices. The level of shielding of the system's components depends on the environment the system is intended to be operated in.

1. Shielding of:
   (a) Cables from signal source to ADC
   (b) External voltage dividers / pre-scalers
   (c) ADC units
   (d) (Site-) Central control computer
2. Distance to strong magnetic field sources (currents)

5.7.3 Galvanic Isolation

Galvanic isolation of all connections to and from the system allows deployment at floating or high potential. It also avoids ground loop problems.

1. Analog inputs (very difficult)
2. GPS receiver
3. Digital I/O lines, event inputs
4. Distributed sampling clock line
5. Network (Ethernet), e.g. optical lines
6. Mains supply
7. Other connections to the system

5.7.4 Overvoltage Protection

Protection against overvoltage of all signal inputs is not essential, but prevents permanent damage in case of accidental wiring mistakes.

1. Overvoltage protection to e.g. 250 V_{ac}

5.8 Maintenance and Design Considerations

The system should be made as easy as possible to design and maintain without sacrificing functionality. This increases the long-term viability of the system.

5.8.1 Software

Factors which affect the design, development, and maintenance of the system's software include:

1. Operating system
   (a) Long-term availability
   (b) Availability of fixes and improvements
   (c) Possibility of maintaining system yourself
   (d) Real-time, general purpose, or general purpose with real-time core
   (e) Availability of programmers (level of specialisation required)
   (f) Runtime stability
2. Portability to multiple platforms
3. Backup facilities
4. Version control system for designing a system
5. Use of data base for archiving

5.8.2 Hardware

Whether assembling a system from commercially available components or designing from the ground up, the following factors regarding the hardware are involved:

1. Vendor support, preferably multiple vendors
2. Specification control
3. Throughput
4. Multi-master capabilities
5. Card size for user hardware
6. Number of slots
7. Method of carrying user-defined signals between user-designed cards (bus system, backplanes)
8. Testing and debugging features

chapter 6

CHART III System Design

6.1 System Components

CHART, which stands for Continuous Harmonic Analysis in Real Time, is a general purpose instrumentation system designed for high resolution data acquisition, precise timing, and real time data processing and analysis [25, 28, 148, 155, 156]. Design of CHART was started because of a need for a system for power quality monitoring, when it was found that existing commercial systems did not fulfil the requirements of continuity, real time, and bandwidth that are described in chapters 1 and 4. A brief overview of CHART is included as an example that fulfils strict accuracy requirements, and to give the technical background for the time base described in chapters 7 and 8 and the tests described in chapter 9.

This chapter describes the third implementation of CHART. This re-development was necessitated by CHART II relying on externally sourced proprietary hardware which was discontinued by its manufacturer. This version, CHART III, uses a standardised I/O bus, the MIX bus [82, 84, 89], for interfacing the CHART proprietary hardware. CHART III was designed to be suitable for mass production, with the intention of turning it into a commercial product. While this commercialisation did not occur, several systems were built for sympathetic customers, as pre-production units. A user and technical reference manual were produced [11, 12].

Since CHART III was developed, aspects of the design have been superseded by technological advances, and other aspects could now be designed in a simpler or more cost-effective way. These advances and new design opportunities are discussed in chapter 11.

An overview of CHART III is given in figure 6.1, with a block diagram in figure 6.2. They show the interconnection of the various parts of CHART III that are explained below.

6.1.1 The Parallel Processing Unit (PPU)

All data is processed and stored in the Parallel Processing Unit (PPU). The PPU is built out of a Multibus II system [68, 80, 81, 83, 85–87]. The Multibus II is a standard designed by Intel which defines a 32-bit bus both electrically and mechanically. A backplane mounted in a rack is used for electrical connection between cards. A 96-pin DIN connector called P1 connects each card to the backplane. This backplane carries the PSB (Parallel System Bus) of the Multibus II. A second 96-pin DIN connector designated P2 has also been made available for user-defined signals. The use of P2 in CHART is described in section 6.2.2.

Mass storage and tape backup devices are connected to the main processor board (HUB) via a SCSI bus available on P2 of the processor board. This SCSI bus is not connected to the P2 connector of any other board in the rack.

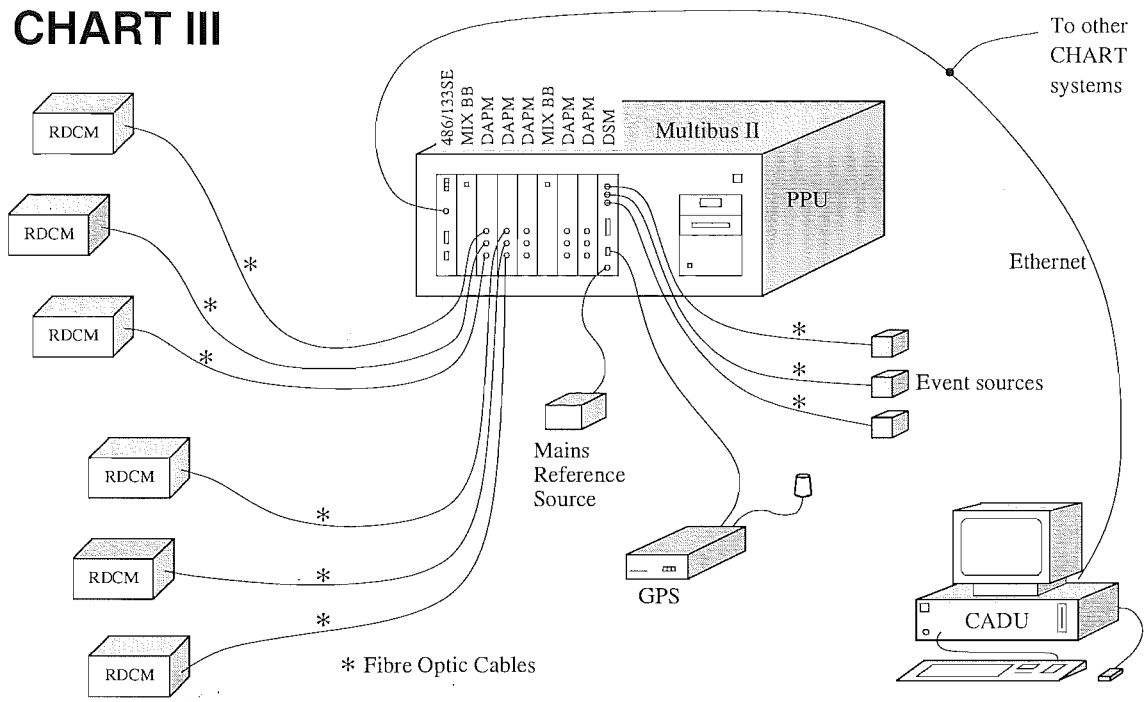


Figure 6.1: CHART III system overview. The Multibus II system shows one main processor card (486/133SE), and two MIX baseboards with three modules each.

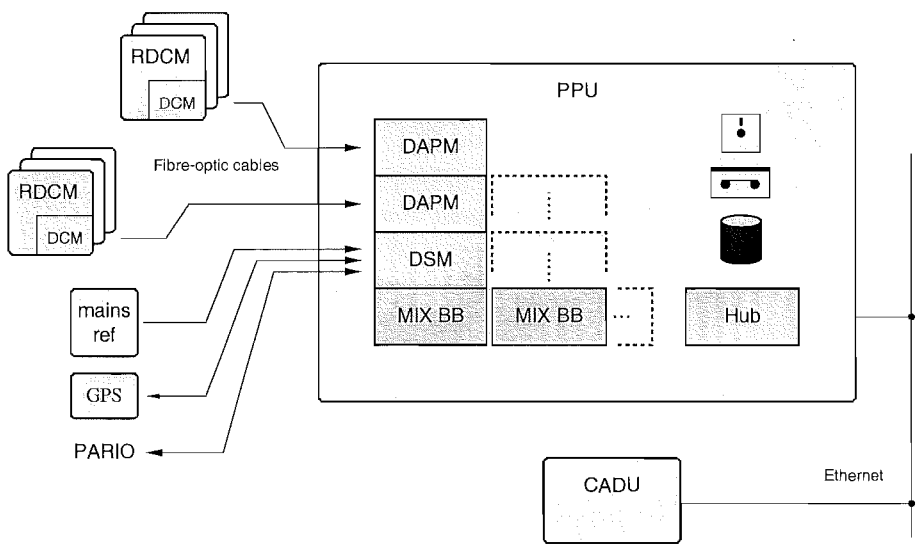


Figure 6.2: A block diagram and overview of the CHART system. RDCM = remote data capture and processing module, MIX BB = MIX baseboard, DAPM = data acquisition and processing module, DSM = digital services module, PPU = parallel processing unit, CADU = control and display unit. The MIX modules DAPM and DSM are not connected directly to the PSB, but are connected via the MIX bus on the MIX baseboards (figure 6.3).

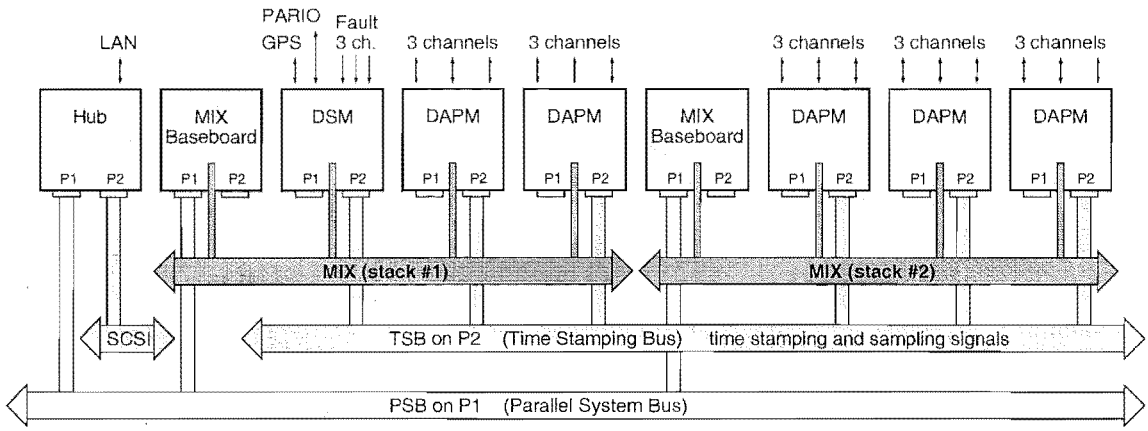


Figure 6.3: Data flow in the PPU. The HUB is an Intel 486/133SE board, The MIX baseboards are Intel 486/020A boards. Hard disk, floppy disk and tape drives are connected to the SCSI bus. The PSB (Parallel System Bus) is not connected to the MIX modules (DAPM, DSM) because they are connected via the MIX bus.

The two custom-designed boards for CHART occupy a slot in the rack but are not logically connected to the PSB. Instead they are interfaced to another card called MIX baseboard. MIX stands for Modular Interface eXtension and is an Intel standard for interfacing custom hardware to a Multibus II system via a standardised interface. See section 7.2 for further details about the MIX.

6.1.2 The Remote Data Capturing Module (RDCM)

The RDCM (Remote Data Capturing Module) converts an analog signal into digital data. It is housed in a weatherproof, shielded case. Power can be supplied by connecting the mains, or by integrated batteries. A fibre-optic cable connects the RDCM with the PPU. A fibre-optic connection protects the signal from strong electromagnetic noise in the environment (e.g. a substation), a situation in which CHART was primarily intended to be used in.

The input stage has variable gain which allows for a large dynamic range of the signals to be digitised. Signals can range from $< 1\,V$ for current probes or clips to $> 100\,V$ for voltage transformers.

The input stage, ADC, and F/O (fibre-optic) interface are fitted into a separate, shielded box called DCM (Data Capturing Module). The DCM can be operated as bench-top unit, power can be supplied by any low-voltage transformer.

The DCM together with a mains voltage transformer, a set of batteries, and a weatherproof case make up the RDCM. Photographs of the DCM and RDCM are shown on page 251.

Each DCM/RDCM occupies one channel of the processing unit. In CHART III, each DCM contains only one ADC. It would be possible to build DCMs with a differing number of ADCs, as well as other speeds and precisions.

6.1.3 The Data Acquisition and Processing Module (DAPM)

CHART III's DAPM (Data Acquisition and Processing Module) is a custom designed board which is plugged into the parallel processing unit (section 6.1.1). It provides three independent channels for processing data captured by DCMs. Each channel is equipped with a fast digital signal processor.

Data is collected and simultaneously analysed by the DAPMs, which reduces the amount of data produced. The data can then be transferred to and stored on the hard disk, which was generously-sized by the standards of the time. The DAPM was designed so that programs for analysing the sampled data could be loaded into the DAPM at any time, thus allowing for an arbitrary number of different ways of data analysis. Programs for harmonic analysis were completed. Users or

customers who have the development tools for the DSP available would be able to program their own data analysis software.

A photograph of the DAPM is on page 252 in appendix H.

6.1.4 The Digital Services Module (DSM)

The DSM (Digital Services Module) is a custom designed board which is plugged into the PPU (section 6.1.1) in a similar manner to the DAPM. The DSM maintains a precision time, generates a system-wide sampling clock, and provides auxiliary digital I/O lines. Systems with multiple DSMs are possible, but care has to be taken — see section 7.9. The design of the DSM is described in detail in the next chapter, and improvements to the DSM are given in chapter 8. This section provides a summary. Similar information can be found in the CHART documentation [103,105].

The global positioning system, which is explained in section 2.2.1, is used as basic time reference by the DSM. A GPS receiver is connected to the DSM via a special cable that interfaces the brand-specific details of the receiver to the interface of the DSM. The GPS receiver used by CHART III itself needs an antenna for its operation.

The DSM is able to generate a sample clock which is synchronised to the frequency of the mains. For this a reference to the mains is required and can be provided via a low-voltage transformer and a connector on the front panel. The clock is generated by a circuit in the DSM called Sample Rate Multiplier (SRM).

Both external events and the internally generated sample clock can be time stamped with the accuracy of the precise time reference maintained by the DSM. Up to three external event sources can be connected to each DSM via a fibre-optic interface, or via the parallel I/O interface. An event source can be anything generating a pulse. The term "event" is used for any signal fed into the event capture channels of the DSM. The event is said to have occurred at the time of the signal edge. The event capture channels are bi-directional, data can be sent from the DSM to the event sources. This is useful e.g. to set up parameters in the event sources.

A number of digital I/O lines are available to the user via the PARIO (parallel I/O) interface. The interface is located on the DSM front panel, and can be controlled by the DSM firmware. The event capture channels can also be accessed via the PARIO interface, which does not require any optical hardware.

Photographs of the DSM are on pages 249 and 250 in appendix H.

6.1.5 The Control and Display Unit (CADU)

The CADU (Control And Display Unit) is the user interface of CHART. It is connected to the PPU via Ethernet. DAPMs and DSMs can be started, stopped, and programs can be uploaded to them. All DSM parameters are set up via the CADU. Stored, analysed sample data can be retrieved from the PPU and displayed. Time recovery results and the measured mains frequency from the DSM can also be displayed.

A photograph showing the time and mains frequency display together with some harmonics displays is appendix H. No further information on the CADU is given because the focus of this thesis is on the design of the DSM.

6.2 System Integration Aspects

Before the work on the DSM was started, the Multibus II architecture[1] had been chosen as the platform for CHART. The bus system is an open standard, although this is not totally true for the

[1] Multibus II is no longer in common use

MIX bus which is used in CHART III as local bus for interfacing the proprietary CHART hardware. The prime constraint on the Multibus II architecture is that Int$_e$l has retained the manufacturing rights for the Multibus II MIX baseboards, but allows any vendor to manufacture the plug-on MIX modules. The MIX standard is completely published, for both the baseboards and the modules [89].

The operating system used on the DSM is iRMX from Int$_e$l. The decision to use this operating system was made in the early stages of CHART because of its real-time support. By the end of the design of CHART III, shortcomings in this operating system in comparison with more modern OSes had become apparent. It is very proprietary, it is not modelled after any existing OS which makes it difficult to become familiar with, it has serious problems handling mass storage hardware, it has essentially no utilities, and support is somewhat limited. In view of these problems, alternative bus systems, available at the time, were investigated and are summarised below:

ISA	Not feasible because of limited bandwidth, and other reasons.
EISA	Might be an alternative but support seems to be fading. Card size possibly too small.
PCI	Is meant to be a local I/O bus, not a system bus. (It is comparable to the MIX bus, not the Multibus).
Micro Channel	Might be an alternative. Card size possibly too small [146].
SBus	Might be an alternative. Proprietary, by Sun Microsystems. Card sizes are possibly small, and it has to be established whether the real-time capabilities of UNIX V are sufficient. UNIX would be a well-supported and widely accepted OS [210].
VME	Might be an alternative. Sufficient card space. Real-time UNIX versions are available.
Futurebus+	Might be an alternative. Sufficient card space.

It was concluded that at that stage no significantly better alternative existed than Multibus II and iRMX. Advances in microprocessors in the last few years have both reduced the real-time OS requirements and improved the commercially available software. This is discussed in detail in chapter 11.

The Global Positioning System (GPS) was chosen as the basic time source for the DSM because it was (and still is) the most cost efficient solution offering an accuracy of better than 1 µs, other than atomic clocks.

6.2.1 Coherent Sampling and the SRM

To implement coherent sampling and to synchronise the analog-to-digital converters (ADCs) in the RDCMS/DCMS, the CHART III system generates a master sampling clock and supplies this clock to all ADCs. This master sampling clock is generated by the Sample Rate Multiplier (SRM) in the DSM, and distributed over the Time Stamping Bus (TSB, section 6.2.2) to the DAPMS. The DAPMS send the sample clock (in all schematics called SAMPLES) via the fibre-optic link to the RDCMS/DCMS, as depicted in figure 6.4. The DAPMS receive the sampled data over the same link.

The CHART III DCMS can handle a sampling rate of up to 200 kHz, and provision was made in CHART for sampling rates of up to 1 MHz. At this frequency the delay of the sample clock from the DSM to the DAPMS is negligible, but the delay from the DAPMS to the DCMS and back can exceed the duration of a sample clock period, depending on the length of the fibre. However, if the lengths of the fibres on all channels are the same, delays are equal and sampling remains coherent.

When computing harmonics of power systems, rectangular windowing by sampling over full periods of the mains fundamental frequency eliminates spectral leakage problems when computing Fourier transforms [148]. To address this the SRM was made capable of producing a fixed, user-defined number of sample clock pulses per mains cycle.

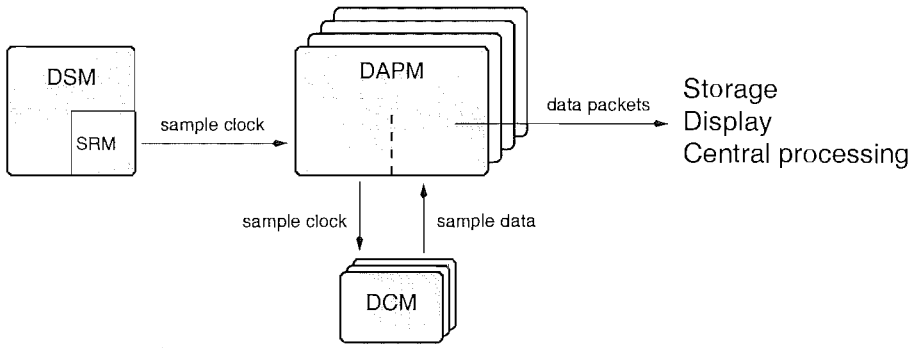


Figure 6.4: The path of the sample clock from the DSM to the DCM, and the data from the DCM to its final destination. Only one path is shown, there can be 3 DCMs per DAPM.

6.2.2 Distributing a Precision Time Throughout the System

The DSM maintains a real-time clock holding a precision time, which has to be transferred to the DAPMs for timing of the acquired data. At accuracies of 1 µs and better, designing this transfer was not trivial. Any software solution would have been far too slow. Previous CHART releases used a counter on the DSM to hold the precision time, and transferred this counter's clock to another equal counter on each DAPM. This solution was deemed not satisfactory by the CHART III design team because if the counters lost synchronisation for any reason correct timing was lost. This concern existed especially for the case when a DAPM was stopped, or when reloading the DAPM software.

In CHART III the precision time is distributed over a parallel bus, the time stamping bus (TSB). Physically the TSB utilises the P2 bus of the Multibus II, which is user-defined. Also part of the 32 bit TSB are the seconds. The coarse time (seconds, minutes, ...) is propagated by software. Each DAPM then matches the coarse time with the precision time using the second information which is part of the precision time.

Any method of transferring time from one card to another through hardware required some sort of bus system. It is possible to wire ribbon cables along the front panels, or through the rack on the inside after solving the problem of inserting the cards with the cable attached, but this is messy. A convenient and tidy way of interconnecting cards is to use backplanes. The Multibus II provides such an access to a backplane with an abundance of free lines.

6.3 Consideration of Timing Errors

The standard design configuration for CHART is shown in figure 6.5. For fault location purposes, a loop antenna is used for measuring $\frac{di}{dt}$ of a power transmission line current. The signal is pre-amplified (PA) and compared with a preset value (TCOMP). If the threshold value is exceeded a signal is carried over a fibre-optic link to the DSM. If two of these systems are set up at two different locations on a transmission line, faults can be detected using the wavefront arrival method [101]. A GPS receiver provides a time reference for each CHART system.

Timing errors introduced by the loop antenna (LA), pre-amplifier (PA), and threshold comparator (TCOMP) are composed of the length of the coaxial cable running down the pylon, and phase shifts introduced at various stages. Signals are carried over conductors approximately at the speed of light[2], the delay is about 3.3 ns/m. For a length of 20 m this results in a delay of 66 ns. Studies have shown that an upper limit of 10 MHz for the bandwidth of these stages is sufficient [101]. A phase shift of 180° at 10 MHz produces a delay of 50 ns. The delay increases for lower frequencies.

[2]Speed of light: $c = 299\,792\,458\ \frac{\mathrm{m}}{\mathrm{s}}$, equivalent to $3.34\ \frac{\mathrm{ns}}{\mathrm{m}}$

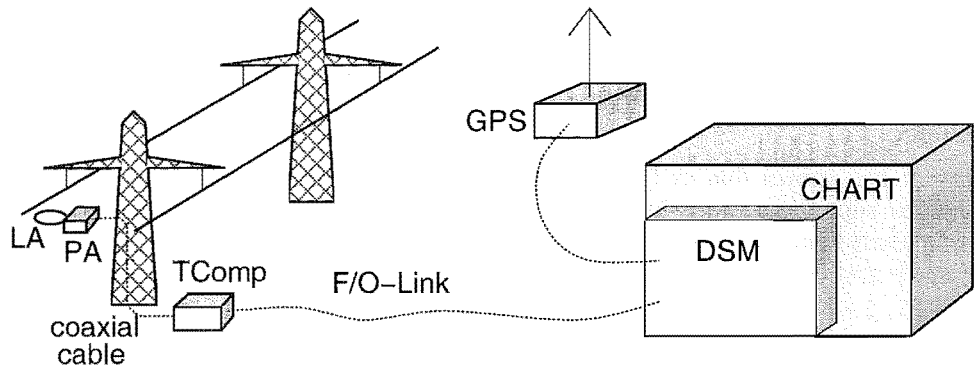


Figure 6.5: Accumulation of time errors in a typical CHART system configuration.
LA = loop antenna, PA = pre-amplifier, TCOMP = threshold comparator.

The delay of the fault signal over the fibre-optic line is given by the length of the line and the signal travelling at (approximately) the speed of light.

The error of the GPS time reference via the `1pps` (one pulse per second) signal has to be added to all error calculations. It is random and can not be compensated for. Three receivers were accessed for use with CHART III. A Trimble receiver that was considered most viable for CHART [219] has an error of ±1 µs, a Motorola receiver [163,164] ±55 ns rms (without SA), and a Magnavox [132–134] ±50 ns rms. Low-cost receivers today are likely to have an accuracy of a few hundred nanoseconds.

On the DSM itself the delays introduced by signal travelling are negligible. The accuracy of the time stamping is ±100 ns, plus an additional delay of 100 ns to 200 ns, as explained in section 7.8.2.

Summing up all errors, the total worst case error is

$$e = e_s + e_t + e_{ts}$$

with
- e total error
- e_s error of the sensor equipment
- e_t error of the transmission line between the sensor and CHART
- e_{ts} error of the time stamping in CHART

The actual total error largely depend on the particular system configuration and the application involved. In some situations errors compensate for each other. For example, if two systems as shown in figure 6.5 are used for fault location, the delays introduced by the sensor equipment are equal. For calculating the fault position using the wavefront arrival method (section 3.4) a constant delay has no effect, as long as it is equal on both measuring locations. This is also the case with the run-time delays of the fibre-optic transmission line. The cable can be arbitrarily long as long as it has the same length at each measuring location.

Previous research [193] has shown that adequate fault location can be performed if the surge wavefronts are timed with an error of less than ±1 µs. This enables faults to be located to the nearest transmission tower or pole. With the DSM introducing a maximum error of +300/−100 ns and sensor and transmission delays cancelling each other out, the GPS time reference may have an error of up to ±700 ns (or, to be precise, up to +700/−900 ns).

6.4 Conclusion

The CHART III system is an example of a high resolution data acquisition system with real-time data processing capabilities, designed to time stamp samples with an accuracy of approximately 1 µs. While it is a general purpose system and could be used for any application requiring these characteristics, it has been specifically applied to power quality monitoring. It comprises a number of RDCMS (remote data capture and processing modules), connected to a PPU (parallel processing

unit) which contains DAPMs (data acquisition and processing modules) and a DSM (digital services module, the time base). Modules on the PPU are connected through Multibus II boards. The time signal is provided by a GPS receiver, and the system can be controlled from a CADU (control and display unit).

This chapter has shown that at the time CHART III was designed, most functionality had to be performed by separate custom-designed hardware modules. Advances in technology since that time, particularly in microprocessor speed, mean that an equivalent system could now be designed making much greater use of off-the-shelf componentry. Such design alternatives are discussed in chapter 11.

chapter 7

The Design of the Time Base

7.1 Overview

This chapter describes the time base that was designed for CHART III, of which the author was the main developer. The time base, the Digital Services Module (DSM), provides the timing information required by the CHART system: it maintains a precise time which is synchronised to the GPS 1pps signal and propagates this time signal to other parts of the CHART III system. The time stamping accuracy is good enough to meet the most stringent power monitoring application requirements, as discussed in chapter 4.

The time base uses a Texas Instruments TMS320C31 Digital Signal Processor (DSP) as described in section 7.3. Improvements in microprocessors which have occured since then mean that an off-the-shelf microprocessor can now be used instead of a specialised DSP as a time base in many applications. This is discussed in detail in chapter 11.

A diagram which shows the data flow of the CHART III time base is given in figure 7.1, and the block diagram showing the implemented hardware is given in figure 7.2.

Interfaces between the DSM and the user's hardware are shown on the left side of figure 7.1. The 50 Hz reference is fed into the zero-crossing detector which is explained in section 7.8.1. The GPS receiver has a bidirectional data connection to the CPU (section 7.4) and a precision time reference signal that is connected to the capture unit (section 7.8.2). Up to three event sources can be connected to the capture unit via fibre-optic interfaces (section 7.5). A low-speed data connection exists from the CPU to the event sources which can for example be used to set up parameters. A number of digital inputs and outputs are provided via the PARIO connector described in section 7.6. This connector also makes the three event capture inputs available and bypasses the fibre-optic interface. All the above mentioned interfaces are available via connectors on the DSM front panel.

The DSM maintains a real-time clock (RTC) with the help of the global positioning system (GPS). The GPS satellite receiver (section 2.2.1) outputs a pulse at the beginning of each second, and, prior to this, date and time via a serial interface. The accuracy of the second pulse is better than 200 ns, although some of this accuracy is lost with the DSM's synchronisation of its RTC to this second pulse. The loss of accuracy is a maximum +300/−100 ns, +200 ns of which is determinate.

By enabling the sampling clock synchronously to the GPS's second pulse, high accuracy time stamping can be achieved. The time at which sampling is enabled falls at the beginning of a second. Higher resolutions than 1 s (for enabling the sampling clock) would be possible by using a frequency-multiplier with the DSM's RTC, but for practical purposes the benefits do not justify the additional hardware costs.

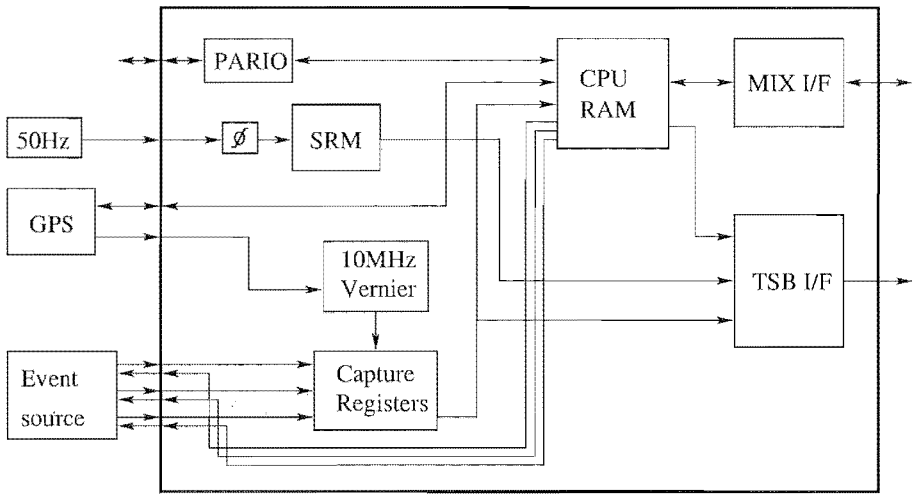


Figure 7.1: DSM data flow.

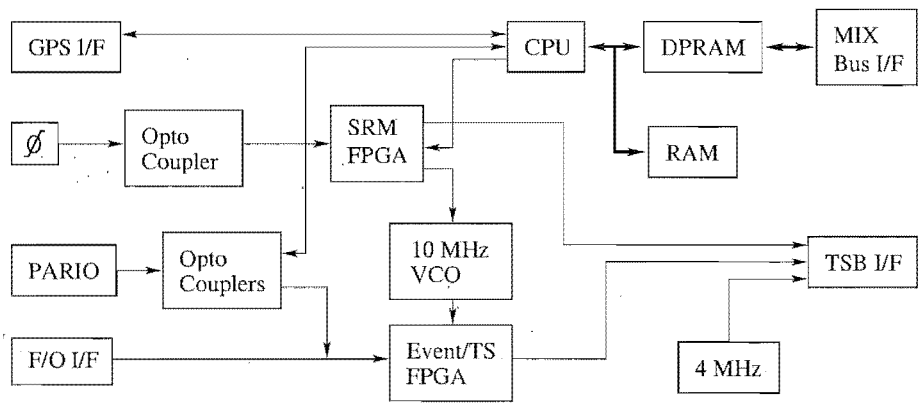


Figure 7.2: DSM block diagram.

The data packets assembled by the DAPM contain in the header the date and time of the first sample in the packet [25]. The resolution of this time stamp is 1 s, the accuracy is in the order of 0.5 µs. This high accuracy is achieved by the central control of the sampling clock.

The accurate time information is distributed via the Time Stamping Bus (TSB) from the DSM to the Data Acquisition and Processing Modules (DAPMS). This is further explained in section 7.7. The DSM interfaces to a standard hardware platform called Modular Interface eXtension (MIX) which is explained further down in section 7.2.

The block diagram in figure 7.2 gives an overview of the implemented DSM hardware. All essential logic has been implemented in two Field Programmable Gate Arrays (FPGAs). Section 7.8 describes both FPGAs in this particular implementation. All functional blocks are described in detail in the following sections.

7.2 Modular Interface eXtension (MIX)

The Modular Interface eXtension [82, 84, 89] is a 32 bit I/O bus with multi-master and interrupt capabilities. It is standardised by Int$_e$l. Up to 3 "MIX Modules" can be stacked onto a "MIX Baseboard", connected with a specially designed 132-pin connector (sketched in figure 7.3). Half-size and full-size modules are possible. A module which can act as bus master is called a master module, a module without these capabilities is called a slave module.

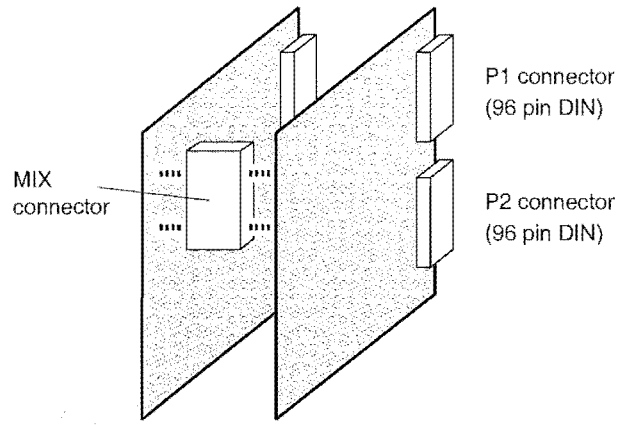


Figure 7.3: A MIX baseboard (right) and a full-size MIX module attached with a MIX connector.

14	09	13	09	10	09	00	09	00	09	00	09	43	09	33	09
31	09	5F	09	44	09	53	09	4D	09	5F	09	5F	09	41	09
01	29	01	09	00	09	00	09	00	09	F0	09	29	09	00	09
00	09	00	09	00	09	00	09	01	29	00	29	00	29	00	29
00	29	01	09	08	09	04	09	08	09	08	09	00	09	08	09
D7	09	07	09	C0	09	00	09	00	09	04	09	C0	09	04	09
17	09	0B	09	00	09	00	09	00	09	00	09	00	09	00	09
00	09	00	09	00	09	00	09	00	09	00	09	00	09	00	09

Table 7.1: Contents of Multibus II interconnect EEPROM. The module identification string C31_DSM__A can be seen.

The MIX baseboard was designed as a Multibus II card. Multibus II is a bus system originally developed by Intel [80, 81, 85, 86, 187], and standardised as IEEE 1296-1987 [68]. It has also been supported by Siemens, Concurrent Technologies, and others, but is no longer in common use now. Multibus II cards use one 96-pin DIN connector and a common back plane. A second 96-pin DIN connector named P2 can be utilised by each card. This connector is user-definable. In CHART it has been used to transfer timing information (see section 7.7). Because only full-size MIX modules have access to the P2 connector, both the DSM and DAPM were designed as full-size modules. Also, the increased board space offered was needed for the DAPMS.

The mechanical design was fairly straightforward, because the geometry of a Multibus II card is standardised, and the position of the MIX connector and mounting elements is fixed. An example for a slave module was provided [82]. Because CHART III did not require vast quantities of data transferred from or to the DSM, the design of the DSM as a slave module was adequate.

Bus drivers were required for both the address and data bus because of line lengths and loading. The address bus drivers had to be inverting. Adequate termination of the MIX bus as specified [89] was required (series 33 Ω resistors). A serial EEPROM holds information about the DSM MIX module for the Multibus II interconnect space (contents listed in table 7.1). The address decoding for the MIX interface has been implemented in a Generic Array Logic (GAL) [117]. A GAL is a programmable logic device which uses E^2 (electrically erasable) cells to store the connect information, this means that the programming of the chip can be erased within a very short time (20 ms). It is also very convenient for development and prototyping. The logic equations for this GAL are listed in appendix D.1.

7.3 The TMS320C31 Digital Signal Processor

The TMS320C31, abbreviated to C31, is a simple, fast 32 bit digital signal processor (DSP) from Texas Instruments [214, 215]. The CPU core has the following features integrated: floating point

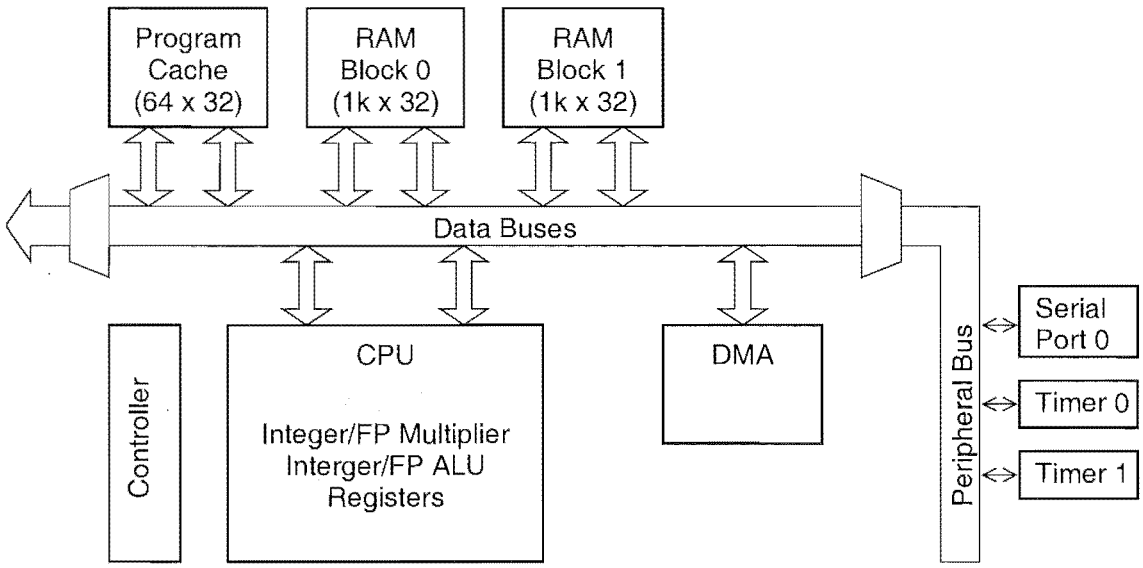


Figure 7.4: Block diagram of the C31 DSP.

arithmetic, cache, 2 RAM blocks, DMA controller, interrupt controller, serial port, and 2 timers. A block diagram is given in figure 7.4.

The C31 can be clocked at 33 MHz resulting in 33.3 MFLOPS and 16.7 MIPS. Single cycle instruction execution time is 60 ns. Operands always have to be 32 bit in size, and no machine instructions are provided for smaller operands. Misaligned bus accesses can not occur. The interrupt controller is comparatively primitive, 4 levels are distinguished and prioritised but an interrupt service routine (ISR) itself can not be interrupted.

The circuitry of the DSP, the RAM, the dual-port RAM, and the MIX interface is essentially the same as for each channel of the DAPM. A green front panel LED is connected to external flag XF0. This flag can be made an output, and the LED can be used to indicate the status under software control, which has been useful for debugging.

Both the RAM and the dual-port RAM are 32 bit wide, thus requiring 4 chips each. The RAM is built with static RAMs with a standard pinout and an access time of 20 ns. The RAMs could be accessed with 0 wait-states but the DUART (section 7.4) is not fast enough for this. For reasons of simplicity the maximum number of wait-states (7) was selected (default), downgrading the performance has not caused a problem because of the high speed of the DSP. The dual-port RAMs are CY7B138 from Cypress [35]. Each of them offers 8 semaphores which are addressed by A2–A0. A semaphore value is written using D0, but when read appears on D7–D0. Semaphores are used as a flag to control access to shared resources, like memory or I/O devices. In the DSM and DAPM they are used for passing messages between a MIX module and a MIX baseboard.

The C31 provides an interface which allows the connection of an external emulator via a 12-pin header connector [213]. All processor functions can be controlled through this interface, and user programs can be down-loaded. The processor in the target system runs the user code under the control of the emulator. This made developing software very efficient — the debugging environment is identical to the target environment because debugging is done on the target system. As well register contents and memory can be examined.

No ROM has been provided for program storage. Instead, the RAM is used for both the application program and the application data. The code is uploaded by the MIX baseboard CPU via the dual-port RAM. The MIX baseboard controls the reset line of the DSP on the DSM, and the dual-port RAM is placed in the DSP address space from where the reset and other vectors are loaded. This allows the user to upload different application programs which was considered to be an important feature for a general-purpose data acquisition tool. It also eliminated the difficulties of handling ROMs during production and made updates very easy to implement.

7.4 The GPS Receiver Interface

The DSM uses the Global Positioning System (GPS) as basic time source. It was uneconomic not to purchase a commercially available receiver. The DSM was designed to fit any of the following receivers:

1. MX4200 from Magnavox [132–136, 140–142].
2. SixGun from Motorola [163, 164].
3. SVeeSix from Trimble [219].

A serial interface is provided for data communication, built with the SCC2692 Dual Asynchronous Receiver/Transmitter (DUART) from Philips [177]. Two 75176 [212] are used as driver/receiver and for conversion to RS-422. All the above commercial GPS receivers were available with an RS-422 interface. The second serial channel of the SCC2692 is used for the fibre-optic interface — refer to section 7.5. The `1pps` signal is connected to both FPGAs (section 7.8). To protect the FPGA inputs from damage a simple clamping circuit is provided with a zener diode. The +12 V line for the power supply is protected with a 500 mA fuse.

A DB-9 connector is attached to the front panel. This connector accommodates all the above signals and has the same pinout as the Trimble GPS receiver [219]. A Trimble receiver can therefore be connected directly, special interface cables are required for other receiver types. A schematic for the Magnavox MX4200 interface cable used for CHART is shown in appendix C.1 on page 216.

7.5 The Fibre-Optic Interface

After a long period of evaluation fibre-optic transmitters and receivers from AT&T were chosen over those from Siemens [199, 200]. They were cost effective and their availability was reasonable, but most importantly they have a small case as narrow as DIP-16, which was important for the densely packed DAPM.

The ODL70 II devices [10] were used for the DAPM. Data transfer rates of up to 70 Mbit/s are possible (in ac-coupled mode), providing a reasonable speed margin for future developments. An ECL interface directly connects to the ECL interface of the TAXI chips [4]. The TAXI chips provide a high-level parallel microprocessor interface as a front-end to a serial transmission line. Error detection and correction are handled as well.

The principal application for the DSM F/O interface is the capturing of events. This requires a fast transmission line between the event source and the DSM, because any delays will increase the error the event is timed with. Unlike the DCM/DAPM transmission, the transmission of events does not involve transferring any data at all. The event itself, represented by a signal edge, is all which needs to be transferred. Therefore, utilising the TAXI-chips is not necessary. The ODL50 II devices [9] were used. They are essentially the same as the ODL70 II. The data rate is limited to 50 Mbit/s. More importantly, they were available with a TTL interface thus making the use of an ECL/TTL conversion unnecessary. Power supply filtering and external circuitry to the ODL50 II were designed as required by the specifications. Because of the stationary characteristic of the signals involved the ODL50 II devices are operated in dc-coupled mode.[1]

The receivers are connected to the event capturing FPGA (see section 7.8.2). The transmitters are connected to the second serial channel of the SCC2692 via a 1-to-3 multiplexer, hence the DSP can only transmit data over one fibre-optic channel at any given time. This is not a limitation as this feature was intended for setting up parameters for the event sources, which is not time-critical.

[1] The 70 Mbit/s data transfer rate of the ODL70 devices is only possible in ac-coupled mode. In dc-coupled mode the maximum data rate drops to 50 Mbit/s.

TS[31:0]	Time information; time stamp
SAMPLES	Clock for sampling
10MHz	Vernier clock
4MHz	TAXI clock
$\overline{\text{EVENT}}$	Event occurred on any channel
$\overline{\text{EVT[3:1]}}$	Event occurred on this channel
TSBVALID	TS[31:0] is valid

Table 7.2: The signals of the Time Stamping Bus (TSB). All are generated by the DSM. The DAPMs also connect to this bus, and use most of these signals.

31 24	23 16	15 8	7 0
vernier 23-16	vernier 15-8	vernier 7-0	seconds
vernier 23-16	vernier 15-8	minutes	seconds
vernier 23-16	hours	minutes	seconds
days	hours	minutes	seconds

Table 7.3: The formats of the time stamp transmitted over TS[31:0] of the TSB. The vernier can be switched off byte by byte, and replaced by a value controlled by the DSP. This feature was used to transfer seconds, minutes, hours, and days respectively. For the pinout see appendix C.2.

7.6 The Parallel Input/Output Interface (PARIO)

The parallel I/O interface (PARIO) provides the user with a number of digital input and output lines which can be controlled by application software running either on the DSM, or on the HUB. The number of lines is only limited by the number of pins the connector can provide. A DB-44 connector (with three rows of pins) was found to provide the most pins while taking up the least space in the front panel.

Opto-couplers are provided for all I/O lines [211]. Eight outputs and four inputs can be fitted. None of these has a common line which means that 2 connector pins are required for each I/O line. The 4 inputs take up 3 connector pins each because an alternative connection is provided with a series resistor which is calculated for an LED supply voltage of 5 V.

The event capture inputs require fast opto-couplers with a short delay; 74OL6010 were used [184]. The power supply required is common for all 3 channels because of the lack of connector pins. The inputs are TTL compatible and tied inactive if not connected. Each of these 3 inputs is ORed together with the output from the respective fibre-optic receiver.

Providing a supply voltage on the PARIO connector would defeat the purpose of galvanic isolation, but possible uses of power include additional, external isolation devices (such as opto-couplers), and production testing of the PARIO interface. The solution that was found to this problem was to supply +5 V and GND via two jumpers. The use of this power supply for testing is described in section 7.10. The power supply and the jumpers were an undocumented feature. The board was shipped with the jumpers open.

7.7 The Time Stamping Bus (TSB)

The Time Stamping Bus (TSB) is a 32 bit bus used to transport time information from the DSM to the DAPMs, plus a few other signals. It was implemented using the user-definable connector P2 of the Multibus II standard. A pinout of P2 can be found in appendix C.2 on page 217. A second

Master Serial	After power-up, the FPGA clocks the configuration in from a serial PROM.
Master Parallel	After power-up, the FPGA clocks the configuration in from an external, 8 bit-wide ROM.
Slave Serial	An external device like a microprocessor clocks in the configuration via a serial line.
Slave Parallel	An external device like a microprocessor clocks in the configuration via an 8 bit parallel port.

Table 7.4: Modes in which the FPGA configuration bit pattern can be loaded. In any of the modes, the FPGA will not be operational until the configuration is completely loaded. The development system can be connected to the FPGA via a DIP-8 socket with the same pinout as the serial PROM. A jumper was provided so that the configuration could be loaded by the development system in slave serial mode during development, and in master serial mode during production.

backplane for P2 was needed in the chassis. For reasons of simplicity and cost the same type of backplane as used for the P1 bus was chosen.

The signals of the TSB are listed in table 7.2. `TS[31:0]` carries 32 bit of time information, the format of this time stamp is shown in table 7.3. The vernier can be switched off byte by byte, and replaced by a value controlled by the DSP. All vernier bits are in binary format. Bits 7–0 of the TSB are always controlled by the DSP. The maximum update rate for the values controlled by the DSP is $1\,s^{-1}$. It was decided that the bytes controlled by the DSP carry seconds, minutes, hours, and days, in binary. The number of vernier bytes is controlled by the user.

SAMPLES is generated by the Sample Rate Multiplier (SRM) and synchronises all sampling performed by the analog-to-digital conversion stages (this signal is given to the DCMS/RDCMS by the DAPMS). `10MHz` is the vernier clock as used by the vernier counter of the event capturing circuitry, refer to section 7.8.2 for details. `4MHz` is a master clock for synchronising the TAXI-transmitters on the DAPMS. $\overline{\texttt{EVENT}}$ indicates that an event has occurred on any of the event capturing channels, and $\overline{\texttt{EVT[3:1]}}$ indicates that an event has occurred on this particular channel. TSBVALID indicates valid `TS[31:0]`, the timing is such that this signal can be used to clock registers on the DAPMS which store `TS[31:0]`, allowing for typical propagation delays over the bus. Not all of these signals were used in CHART III, but they were included for future development.

The DAPM combines the time information read from the TSB with the sampled data. Because of bandwidth limitations, this is only done once per assembled data packet, for the first sample in the packet. The time stamps of the remaining samples in the packet are determined by the time of the first sample, and the sampling rate. If, for any reason e.g. fibre-optic transfer errors, samples are missing, or sampling is stopped in the middle of a packet and re-started later, time stamps for the remaining samples in this packet are not available. This situation will cause synchronous sampling in particular to fail.

All the TSB signals are buffered with 74ABT245 octal bus drivers [176] before going to the P2 connector, a pinout of which is shown in appendix C.2, page 217. Their outputs can be disabled. In systems with more than one DSM connected to the same P2 backplane (this is the one carrying the TSB), the output drivers of all but one DSM must be disabled by pulling the respective jumper. *Failure to do so might result in permanent hardware damage.*

7.8 The Field Programmable Gate Arrays

The common problem of circuit complexity and limited board space was apparent in CHART, especially with the DAPM design. Faster development and higher integration density was a requirement that could not be met by standard TTL technology. Programmable logic seemed ideal. In

some cases (like address decoders) the use of Programmable Logic Devices (PLD), or the electrically erasable variety called GAL, is very efficient. However, for more complex circuitry a higher-integrated device is needed. After evaluation the Field Programmable Gate Arrays (FPGA) from XILINX were chosen. A development system for these devices had previously been bought by the university, which was a deciding factor in the choice of XILINX.

A XILINX FPGA [230] consists of an array of Common Logic Blocks (CLB), input and output drivers, 2 global clock buffers, crystal oscillator support, and routing facilities. Each CLB has 5 inputs and offers 2 flip-flops and combinatorial logic. The structure is similar to a logic macro cell found in PLDs. The switching of connections is accomplished with multiplexers, the state of which is stored in static RAM cells. After power-up a configuration bit pattern must be loaded into the chip, various different ways to do this are supported, see table 7.4. The configuration can be re-loaded at any time.

The development system combines a variety of software packages from different vendors, mainly a schematic capture package, and the design compilers and simulators. A top-level user interface program called xdm acts as front-end to the other parts of the software used for various design stages. A make utility runs the design compiler and various conversion programs. Options can be selected via menus.

ViewLogic is integrated as a schematic capture package[2]. ViewSim can be used for a functional simulation of the design. It is easy to use and offers many functions. A full timing analysis can also be performed. Editing on the chip level is possible with EditLCA, which also allows a detailed delay analysis to be performed.

The 3100 series of FPGAs was chosen because of speed requirements, and the amount of logic offered was satisfactory. Late in development, XILINX brought the 3100A series to market which was compatible with the 3100 devices, but the routing resources were increased, and the cost was lowered. Therefore all later acquisitions were made from the 3100A series. Both FPGAs used for the DSM must be 3100A series devices. Because of increased routing resources, the 50 MHz counter implemented in one of the FPGAs was able to run with the required speed, and the complex design for the other FPGA could be fitted. The DSM improvements described in chapter 8 require the 3100A.

From the DSP's point of view both FPGAs were designed as memory mapped I/O devices. Furthermore, both FPGAs appear as one entity to the DSP. The same method was used in the Commodore Amiga computer [34] to simplify software development and to increase flexibility with respect to future hardware improvements. Unused logic that might be inserted into the design of a particular chip with this method would be completely removed by the logic optimiser. This was the case with the address decoder for the FPGAs, for which there is only one design for both FPGAs.

Details about how the configuration of the two DSM FPGAs is loaded can be found in appendix F.

7.8.1 The Sample Rate Multiplier FPGA

The Sample Rate Multiplier (SRM) creates the master sample clock which is supplied to the whole CHART system for coherent sampling. Refer to appendix E for the schematic diagrams. Figure 7.5 shows the FPGA block diagram. A 50 MHz master clock is divided by a counter to generate the sampling clock. The division factor is a 12 bit number, resulting in a minimum sample frequency of 12 207 Hz. A parallel loadable counter was needed for this, and speed required a synchronous counter implementation. The design for this counter was taken from a XILINX application note [229] and modified as needed. Despite the optimisations used in the counter design, only a maximum clock frequency of about 45 MHz could be reached with the 3100 series devices. This was determined by a delay analysis inside the EditLCA program. Delays between any of the Q-outputs of the counter flip-flops and their D-inputs were analysed. The lowest maximum delay that could be achieved was 22.5 ns. The problem was overcome by using a 3100A series device,

[2] The customer has the choice between ViewLogic and OrCAD.

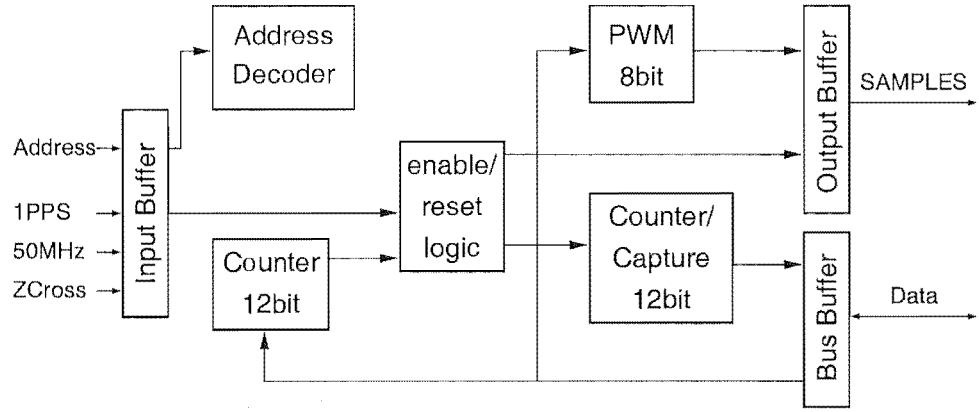


Figure 7.5: Block diagram of the XC3142A SRM FPGA. For detailed schematics see appendix E.

the increased routing capabilities resulted in shorter connections on the chip and pushed the maximum delay down to 12.3 ns, well below the 20 ns required for a 50 MHz clock.

The division ratio is controlled by software, and the control loop is effectively a Phase Locked Loop (PLL). This is the same principle as was used for the previous implementation [101]. To create a sample frequency which provides a preset number of samples per mains cycle, the division ratio is adjusted accordingly after counting the number of samples per mains cycle. This was implemented with a carefully designed counter/capture circuitry, which is described in section 7.8.2. Delays, and flip-flop setup and hold times had to be considered. It was also observed that the capture counter must never be reset as this loses up to one count each cycle, resulting in a "jumping" display. To obtain a stable display it was necessary to have exactly the required number of samples per cycle, averaged over several mains cycles. If, for example, there was half a sample missing in the current cycle, the following cycle must be half a sample longer. If this half sample were merely truncated, the PLL would be fooled into the assumption that the correct number of samples had been generated per cycle, while in reality there would be an average of half a sample per cycle missing, resulting in a display that moved over the time axis. The number of samples generated for the cycle which has just finished had to be obtained by subtracting the previous sample counter value from the current one.

The SAMPLES signal can be inhibited by software. When it is released all DCMS start sampling at the same time. The inhibition of the sample clock can be started or stopped by the `1pps` signal from the GPS receiver, thus making it possible to start or stop sampling at a certain, preset time.

A bidirectional bus interface to the DSP must be implemented in this FPGA because the DSP must read the counter/capture value, and write the division ratio of the 50 MHz divider. A register has been implemented which is controlled by the DSP, providing signals for inhibiting sampling, etc. The address decoder needed for this is the same as for the second FPGA to ensure uniform appearance. Care was taken that, depending on which register is being read, the output buffers of only one FPGA are enabled.

A simple 8 bit pulse width modulator (PWM) used as digital-to-analog converter (DAC) has been implemented into this FPGA as well. It is used for the voltage controlled oscillator (VCO) explained in the next section.

7.8.2 The Time Base FPGA

The main task of this FPGA (figure 7.6) is to maintain accurate timing information and to be able to time stamp captured events. The `1pps` signal from the GPS receiver indicates the beginning of a second with an accuracy of better than 1 µs. This is only useful for time stamping events with a resolution of 1 s. To increase the resolution, a vernier counter is used which is synchronised to

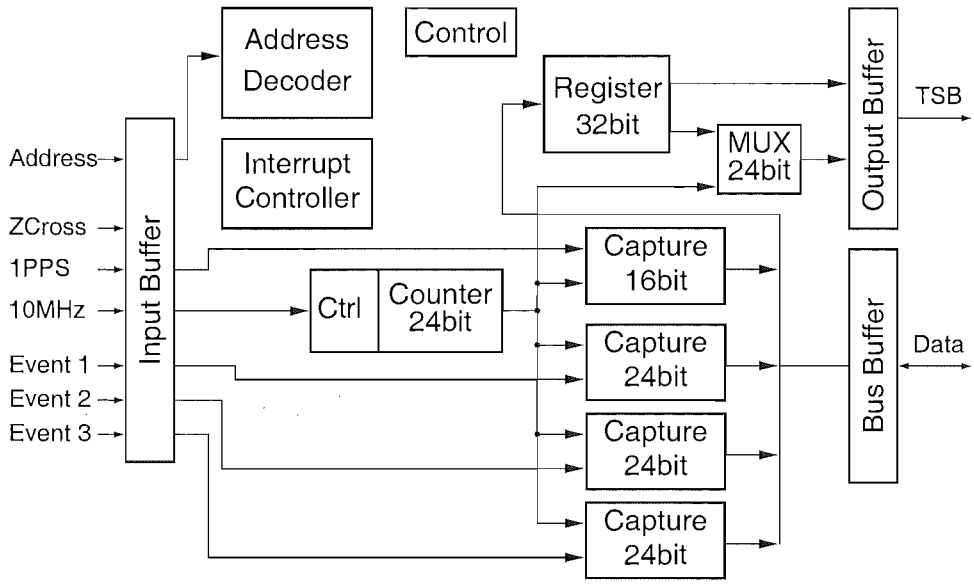


Figure 7.6: Block diagram of the XC3164A time stamping FPGA. For detailed schematics see appendix E.

the `1pps` signal. The number of counts of the vernier counter represents the time elapsed since the beginning of the second. This principle is the same as used previously [101].

The signal `1pps` and any of the events clock the respective capture register, storing the current vernier counter value. To ensure that the counter value is clocked correctly into the registers, it is necessary to synchronise the capture register clock to the vernier counter clock.

Generally speaking, synchronising a signal A to a signal B means clocking A into a D flip-flop, using B as flip-flop clock. The Q output of the flip-flop then represents A, but every change in A is delayed until the next active edge of B. The frequency of A is typically lower than that of B.

Synchronising `1pps` to the vernier clock means that the time reference signal is delayed by up to one vernier clock cycle, or up to 100 ns. Although the total time stamping error is increased by this amount, the synchronous design ensures that there are no potential signal races in the time stamping circuitry. Because of a further synchronisation stage in the TSB bus interface, a second clock cycle is lost, thus delaying the event and `1pps` signals by up to two clock cycles. This is well within the accuracy required for the system.

`1pps` itself can have an accuracy of about ±200 ns. 10 MHz voltage controlled oscillators (VCO) are common, and with this vernier clock frequency the overall accuracy is better than the required 1 µs. The control range for the frequency only has to cover the inaccuracy of the frequency, the frequency is controlled such that the VCO runs at nominal frequency ±1 clock. The vernier counter must count for at least 1 s which is up to 10^7 at 10 MHz, thus requiring a 24 bit binary counter (2^{24} is the smallest power of 2 larger than 10^7).

Each of the three event capture channels has its own capture register. The frequency of the vernier clock is adjusted under software control such that there are exactly the nominal number of counts in one second (in this case, 10^7). This is effectively a phase locked loop as described for the SRM counter. The `1pps` signal is used as a reference. A fourth capture register counts the number of vernier clocks in 1 second, the 1 second given by `1pps`. The control voltage for the VCO is generated by a PWM circuit and a low-pass filter. If the `1pps` signal is unavailable, because for example the GPS receiver is not connected, the value for the control voltage of the VCO in the PWM will not be changed and remains. From then on the time stamping accuracy of the VCO depends on its long-term stability. For this reason a temperature compensated VCO from Rakon was chosen [189].

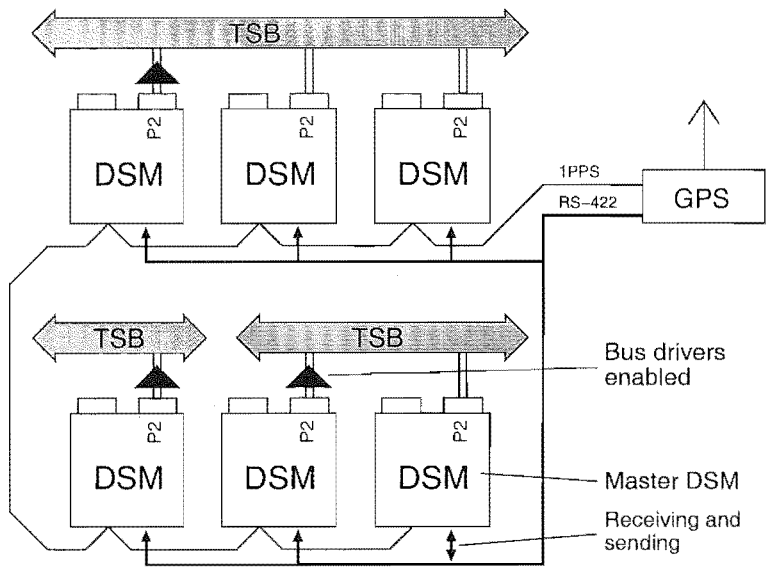


Figure 7.7: Operating more than one DSM in a system. The black triangles indicate enabled TSB drivers. For each separate TSB exactly one DSM must have its TSB drivers enabled. Transmit data (Tx) of the serial connection of the GPS receiver, and `1pps`, are connected to each DSM. Receive data (Rx) of the GPS receiver is connected to only one DSM, this master DSM initialises the GPS receiver.

For practical reasons the vernier counter counts from $2^{24} - 10^7$ to $2^{24} - 1$. The logic blocks used for implementing the counter provide a flag for the state $2^{24} - 1$ (all 1s). Any software handling the vernier time must take care to subtract $2^{24} - 10^7$ from any value.

The capture register for the vernier counts is only 16 bits wide; this register will always be clocked near a 1 s boundary, therefore the higher counter bits will always have the same state as the highest one: 1 if the counter was just before reaching terminal count, and 0 if the counter had already overflown. Special care has to be taken when clocking the capture registers, it can not be done while the counter is counting, to avoid spurious values — see above.

An interrupt controller administering 6 interrupts was implemented: 3 event capture channels, `1pps`, zero-crossing, and vernier counter overflow. The 3 event interrupts can be turned off, and have been assigned to one DSP external interrupt. The remaining three interrupts have been assigned to another DSP external interrupt.

The address decoder is the same as for the SRM FPGA for uniform appearance.

7.9 Multiple DSMs in the Same System

To increase the number of event capturing channels it is possible to have more than one DSM in a Multibus II system as shown in figure 7.7. There can be only one DSM driving a particular Time Stamping Bus (TSB). A Multibus II system is not limited to one TSB, there can be as many as the mechanical structure allows. Multiple DSMs can be connected to the same TSB, but only one of these DSMs can drive it. The TSB drivers of the other DSMs must be disabled by the jumper (J1) provided in the design.

The `1pps` signal from the GPS receiver is connected to each DSM. Maximum line length requirements must be met. The RS-422 serial data connection is also connected to each DSM. The Tx (send data) line from the GPS receiver is connected to the Rx (receive data) line of each DSM, so that all DSMs receive the time output by the receiver. In the opposite direction, only one DSM has its Tx line connected to the Rx line of the GPS receiver (figure 7.7). This master DSM initialises the GPS receiver.

7.10 Considerations for Production and Testing

CHART III was aimed at achieving a pre-production quality product. As a result of this all PCBs had to be designed in a manner suitable for mass-production and fast testing.

The effect on the PCB layout was that sufficient space had to be provided between components so as to allow assembly with pick-and-place machines. Copper rings uncovered by solder stop mask had to be put in certain locations to aid the placing machines.

Test racks were built for on-the-bench testing. Photographs of the test racks are given in appendix H. A laboratory power supply with current limiting was used to power-up the board, and the C31 emulator was required for down-loading the software. Test programs were written to test the memory. A special test program was developed to test the DSM PARIO and event capturing interfaces. Additional hardware that assisted with testing included:

- Serial test cable which connects the serial interface of the GPS receiver with a serial terminal. An MS-DOS computer with serial port and MS-Windows' terminal program was used.
- Adaptor which is plugged in the PARIO connector.
- A fibre-optic loop cable which connects the transmitters with the receivers, for each channel.

The two jumpers which supply 0 V and +5 V to the PARIO connector must be closed for the test adaptor to operate. The test cable needs a ±12 V supply for the internal RS-422-to-RS-232 converter. +12 V is available on the GPS connector, the −12 V must be supplied externally. A more sophisticated design would take the power from the P2 connector where it can be supplied by the test rack.

The progress of the test program is shown on the terminal. The 4 LEDs on the PARIO test adaptor must light up in a certain sequence, failure to do so means faults in the parallel I/O lines. Scrambled characters on the terminal screen mean that the serial interface is not working properly.

7.11 Printed Circuit Board (PCB) Layout

The schematics shown in appendix C and the PCB layout were produced with Protel Schematics and Protel PCB, a package from the Australian software house Protel running under MS-Windows.

The PCB layout had to fulfil the following requirements:

1. Dimensions of a Multibus II board, with connectors, front panel, etc. in the correct places. Components had to be below a maximum height.
2. Full-size MIX module, with the MIX connector and various pieces of mounting hardware, like stand-offs, in the correct locations. All requirements imposed by the MIX standard had to be met.
3. Size and clearances of tracks had to be to the specifications of the PCB manufacturer. As is standard for PCB design, tracks had to join SMD pads in a certain way to prevent the component from being pulled off the pad.
4. The layout had to be suitable for automated assembly of the board. This meant minimum clearances around components, and eyelets next to chips with a narrow pin spacing, such as the C31 CPU, to assist the automatic placing machine with positioning.

The DSM design with the TMS320C31 uses 8 layers, which is the same number as the DAPM uses.

The pin spacing of PLCC cases is 0.05 in, also known as 50 mil (1 mil = 0.001 in). The 50 mil are evenly occupied by the width of the pin and the space between two pins (figure 7.8). With space to either side of a track between two pins, it leaves 8 mil for the width of the track and 8 mil between the track and the adjacent pins (neglecting 1 mil altogether). An 8 mil structure approached the limits of the PCB manufacturer and was regarded as an absolute minimum. Only the DSP chip had a pin spacing of less than 25 mil, making it impossible to place tracks between pins of this chip. This was not a serious limitation because it was possible to have tracks on any of the other layers.

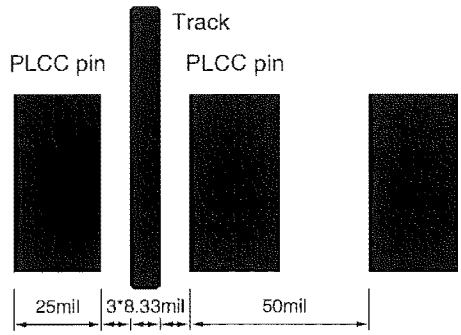


Figure 7.8: Spacing of PLCC pins.

Error Code	Meaning
2	CPU overflow. The total time needed to call the "often per second" and "once per second" routines of each module once exceeded 1 s.
3	Failure to initialise the MIX interface
4	Failure to send a message through the MIX interface
5	Self-test finished (only used by the test program)
6	Failure to configure FPGAs
7	Failure to configure the time stamping FPGA

Table 7.5: DSM error codes displayed by the front panel LED.

7.12 Firmware for the DSM

All programming was done in ANSI-C. Coding for main functional units like sample rate multiplier (SRM) or real-time clock (RTC) was split into separate modules. This allows easy maintenance and conforms to the C programming philosophy. The modules generated were:

dsm	Main module
rtc	Real-time clock
srm	Sample rate multiplier
mix	MIX communication routines (This was based on two modules which are shared with the DAPM)
duart	Serial interface
gps	High-level interface for all GPS receivers.
mx4200	Driver for the Magnavox MX4200 GPS receiver.
testaid	Routines which aid in testing, and software development. Some are also used by the normal program.

Interrupt service routines (ISRs) were implemented for each of the four external interrupts to the CPU. One interrupt is occupied by the message passing software, two by the time stamping circuitry, and one by the DUART.

For software development, a terminal could be connected to DUART channel A (via the GPS serial connector) or channel B (via test clips). Functions implemented in the module testaid (section 7.12.8) output messages or variables. These functions are called from strategic locations within the software, the calls being enclosed in conditional compiler directives. These test output functions are also used by the production test program.

If a fatal condition is detected during normal operation, an error number is displayed continuously and normal operation is halted. The error number is displayed using the front panel LED with the number of flashes representing the error number. A longer pause is inserted before the flashes are repeated. The error numbers used by CHART III are listed in table 7.5.

7.12.1 Module DSM

This module is the main module. It initialises global variables, loads the configuration into the FPGAs, calls the initialisation part of each of the following modules, and enters the main program loop. Some of the software problems of the DSM could be solved best by a small, efficient multi-tasking kernel, as many of the tasks are completely separate from each other but still have to run simultaneously. This programming approach would make all parts easily maintainable and create a consistent logical structure. Unfortunately, such a kernel was not available for the design of CHART III. Alternatives that are now available are discussed in chapter 11.

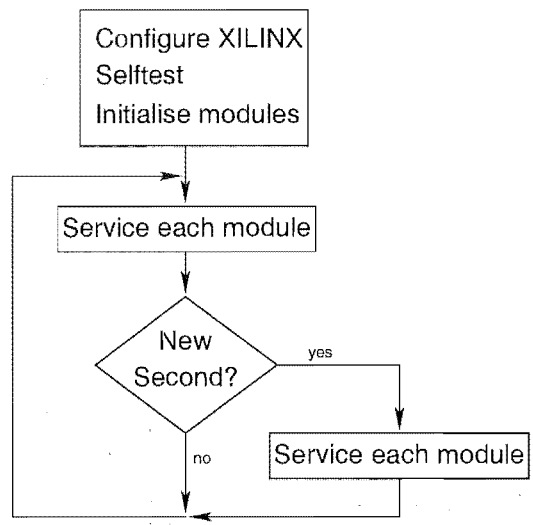


Figure 7.9: Simplified flow diagram of the DSM firmware.

For CHART III, an approach that was sufficient in this case was implemented, shown by figure 7.9, which effectively resembles a simple multi-tasking kernel. Each module must provide two routines which are called periodically by the main loop, simulating multi-tasking. One routine contains all actions that must be performed more than once per second, the other routine contains all actions that must only be performed once per second.

7.12.2 Module RTC

Initialisation
Initialise module variables. Enable SRM/TS (SRM/time stamping) interrupt.

Often per second
Check whether a complete line has been received from the GPS receiver. If yes, a routine from the mx4200 module is called to decode the line and extract the time recovery information. (This information is output by the GPS receiver prior to the start of the second indicated by the `1pps` signal.)

Once per second
These steps are performed:

1. Adjust the frequency of the voltage-controlled crystal oscillator.
2. Transmit the time to the rest of the CHART system by sending it to the MIX interface.
3. If it has changed, transmit the status of the `1pps` signal by sending it to the MIX baseboard.
4. Update the part of the TSB containing the coarse time.

Other

Interface functions for updating the RTC time, obtaining it, and an indicator whether a new second has started were implemented.

The ISR updates the RTC time every second with the time received from the GPS receiver. If no new time is available from the GPS receiver, the RTC time is incremented by software, in which case the accuracy of the RTC is equivalent to the accuracy of the voltage-controlled crystal oscillator.

7.12.3 Module SRM

Initialisation

—

Often per second

—

Once per second

These steps are performed:

1. Detect whether the mains reference signal is provided.
2. Send the measured mains frequency to the MIX interface, if the mains reference signal is currently provided.
3. Check whether sampling is deferred and if so, check whether sampling has to be started or stopped.

Other

Calculate the mains frequency from the reference signal, using a moving average filter.

The ISR adjusts the division factor of the pre-scaler according to the mains frequency and the mode of the SRM. Two modes are implemented:

1. Free running at a constant, user defined frequency.
2. Synchronised to mains frequency with a constant, user defined number of samples per mains cycle.

7.12.4 Module MIX

This module handles the communication with the MIX baseboard, i.e. passing messages between the DSM and the MIX baseboard. It makes use of two other modules, MSG and DPRMSG, which were implemented for the DAPM. The message passing routines of the DSM and the DAPM are the same. Further information about how the message communication was implemented on a lower level can be found in reference [151].

Initialisation

Send messages which are required to establish the communication chain to the CADU.

Often per second

Check whether new control or setup values have been received, if yes, routines of the respective modules are called to update their setup variables.

Once per second

—

Other

Three routines generate appropriate messages to transfer time, measured mains frequency, or the status of the `1pps` signal. This hides the internals of the message handling from the rest of the DSM software.

7.12.5 Module DUART

Initialisation

DUART registers are set up to the communication parameters as obtained by a call to a routine in module mx4200.

Often per second

—

Once per second

—

Other

Converts communication parameters like "2400 8N1" into a bit pattern for setting up the DUART chip to these parameters.

The ISR receives a character from the GPS receiver and stores it in a buffer. If a line is received or the buffer becomes full the contents are copied to a second buffer to make the serial interface immediately available again, and a "received line from GPS receiver" is flagged. The ISR also checks if there is another byte to be transmitted to the GPS receiver, and transmits it if there is.

7.12.6 Module GPS

This module provides a standard interface to all types of GPS receivers. For CHART III only one type (MX4200) was supported, so it consists only of a C header file. This module only provides a switching mechanism, or front end. The particulars for each type of GPS receiver are implemented in a separate module. The following functions must be provided for each GPS receiver type:

1. Get communication parameters. This function returns the parameters for the serial interface of the GPS receiver.
2. Initialise GPS receiver.
3. Convert a line received from the GPS receiver via the serial interface into the time represented by this line. Returns 0 or −1 if the line does not represent a time.

7.12.7 Module MX4200

The main task of this module is to hide the specifics of the Magnavox MX4200 GPS receiver from the rest of the DSM software. It provides a "driver" for this receiver. If multiple GPS receiver types were supported, each receiver type would have one of these modules, which interface to the module GPS. Other DSM modules only interface to module GPS.

Initialisation

Send initialisation strings to the GPS receiver using the `send_string` routine of module duart. Module duart must be initialised before module mx4200.

Once per second

—

Often per second

—

Other

Two other routines return the communication parameters for the receiver, and decode the time from a passed string.

7.12.8 Module TESTAID

This module combines these functions useful for testing and debugging:

1. Turn front panel LED on/off, or toggle it.
2. Wait N milliseconds.
3. Flash the front panel LED in certain intervals to communicate a given number to the operator. (This is used to signal error conditions.)
4. Integer-to-hex conversion. Used for debugging.
5. Test output. Explained below.

Either one of the two serial channels of the DUART can be used for outputting messages. The channel must be opened first, specifying communication parameters. Functions are provided for outputting strings, and 2, 4, and 8-digit hex numbers. Extensions to handle input from the terminal would be possible.

7.12.9 Message Passing

The message passing routines use a customary protocol and the hardware semaphores of the dual-port RAMs for exchanging messages, or data structures, between the MIX baseboard and the MIX modules. The same protocol is implemented throughout CHART [151, 153].

7.13 Conclusion

This chapter has described the hardware design of the DSM (Digital Services Module), which is a precise time base for the CHARTIII instrumentation system. The DSM is a custom-designed hardware module based around a Texas Instruments TMS320C31 DSP. Its main characteristic is the ability to generate signals which are suitable for time stamping sample data with an accuracy of 0.5 µs.

The hardware is controlled by software which is loaded onto the DSM when the instrument is configured for measurement. Because the software is loaded before each use, it is possible to adapt the software for future needs or to load different software for different types of measurements. The same applies to the circuitry which is loaded into the two FPGAs by the DSM software.

The time base makes CHARTIII suitable for distributed measurements that need to compare sample data between geographically separated measurement sites with high accuracy.

chapter 8

Improvements in the Sample Pulse Generation and Time Stamping

8.1 Introduction

This chapter outlines the changes and improvements made by the author to the DSM software and FPGA designs. Its DSP software is written in ANSI-C [216, 217], with an option for the CHART user to implement user applications, as outlined in section 8.8.

Section 8.2.1 summarises the SRM (Sample Rate Multiplier) modes. These control the sampling frequency. Section 8.2.2 explains how the samples are taken, and section 8.2.3 explains when the samples are taken. The SRM mode, sampling mode, and sampling control are independent from each other and can be used in any combination. Examples of how the new sampling features can be used are shown in section 8.2.4.

The remaining sections deal with some of the implementation details. Section 8.3 outlines changes to the handling of FPGA designs. Sections 8.8 and 8.5 explain software changes to the DSM. Section 8.9 introduces new CADU display windows which were required for the functionality described in this chapter but were programmed by another team member.

8.2 Generation of the Sampling Clock Signal

The initial design of the DSM allowed a choice of whether the sample frequency is synchronised to mains frequency or not, but did not have further control over when measurements could commence or finish. Measurements could be started only by pushing the start button on the CADU, and could only be stopped in the same fashion. This is not only unsatisfactory for measurements outside business hours (particularly in the early hours of the morning), but also for more elaborate investigations occurring at regular intervals.

The GPS receiver used to have to be connected and be supplying the `1pps` signal for the sampling to start. Any instrumentation system should be operational without external time reference if the user desires. To allow for this possibility, the DSM's hardware and software were modified.

The following three sections describe enhancements for how the sample clock is generated (SRM mode), and how it is turned on and off (sampling mode and sampling control).

8.2.1 SRM Modes

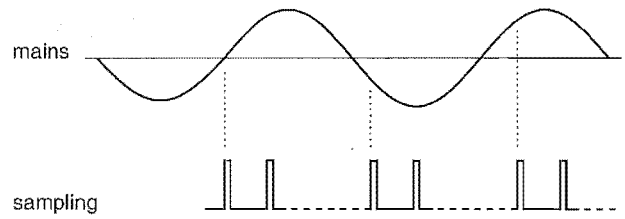
The SRM (sample rate multiplier) mode controls the sampling frequency. The following modes are available:

None

No sampling signal is generated.

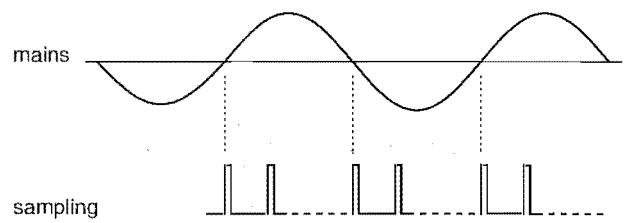
Free running

The sampling frequency is specified by the user. The sample points do not have any relationship with the mains cycle.



Synchronised to mains frequency

The sampling frequency is a multiple of the mains frequency. The user specifies the number of samples per mains cycle. This SRM mode does not establish any known relationship of particular samples with mains phase or mains angle; it only keeps the number of samples per mains cycle constant.



The sampling clock is derived by dividing a high frequency (50 MHz) by a calculated number. The sampling frequency is controlled by adjusting this divisor. Because of this, only discrete sampling frequencies can be generated. The consequences of this are examined in reference [148].

8.2.2 Sampling Modes

The sampling mode determines how the generation of sampling pulses begins and ceases. Timing diagrams for burst and packet sampling are given in appendix G. The following choices are available:

Continuous sampling

Sampling is turned on, and off, at the beginning of a second. The beginning of a second is determined by the RTC (real-time clock) maintained by the DSM, or by the `1pps` signal of the GPS receiver, if available. This means that the DSM is fully operational without GPS receiver as well as with.

Burst sampling

A burst of samples are generated at the beginning of a second. The number of samples can be specified. When the specified number of samples have been taken, sampling is turned off for the rest of the second.

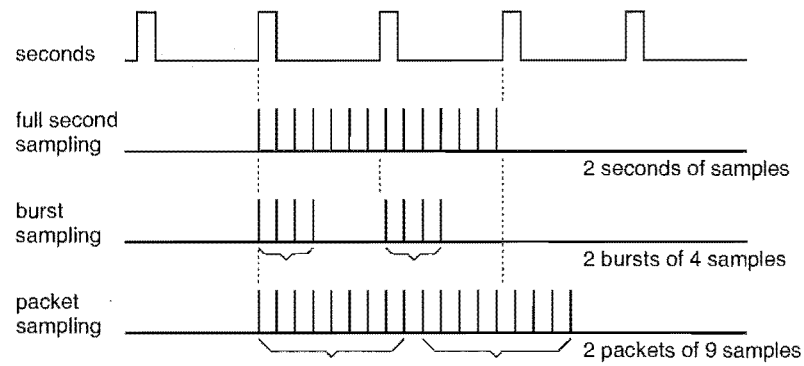
Between 16 and 65536 samples can be generated this way. The number of samples must also be a multiple of 16.

Packet sampling

When sampling is turned on, the total number of samples is counted, and sampling is not turned off before a multiple of the specified number of samples is taken. This ensures that the last packet of samples is completed before sampling ceases. Sampling is turned on at the beginning of a second, and turned off at the beginning of the second following the completion of the last packet.

Between 16 and 65536 samples can be generated this way. The number of samples must also be a multiple of 16.

All sampling modes are illustrated for comparison:



8.2.3 Sampling Control

Sampling control determines when the generation of samples is started or stopped. The following choices are available:

Continuous sampling

Sampling is turned on with the beginning of the next second, and left on until manually stopped. When stopped, sampling is turned off at the beginning of the following second. (This is identical to the operation as before the improvents.)

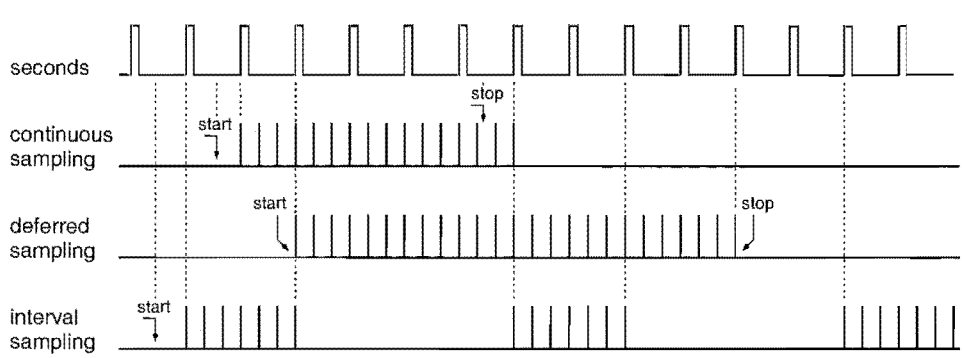
Deferred sampling

Sampling is turned on at the start time, and turned off at the stop time, both at the beginning of the specified second.

Interval sampling

At the beginning of each interval, sampling is turned on for a certain length of time, then turned off for the rest of the interval. This is then repeated. The number of seconds for the interval length and the on time can be specified. Sampling is turned on or off at the beginning of a second. An interval length of 0 s turns interval sampling off.

The different methods of sampling control are illustrated:

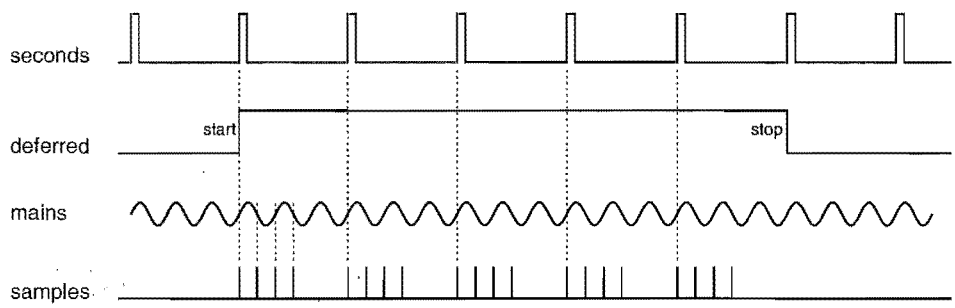


As shown on the diagram above, when sampling continuously, sampling is turned on and off at the beginning of the second following the one in which it has been started or stopped. In deferred mode, sampling is turned on at the beginning of the specified second, and turned off at the beginning of the specified second. In interval mode, sampling is turned on for the specified number of seconds at the beginning of each interval (here: interval length 6 s, on time 2 s).

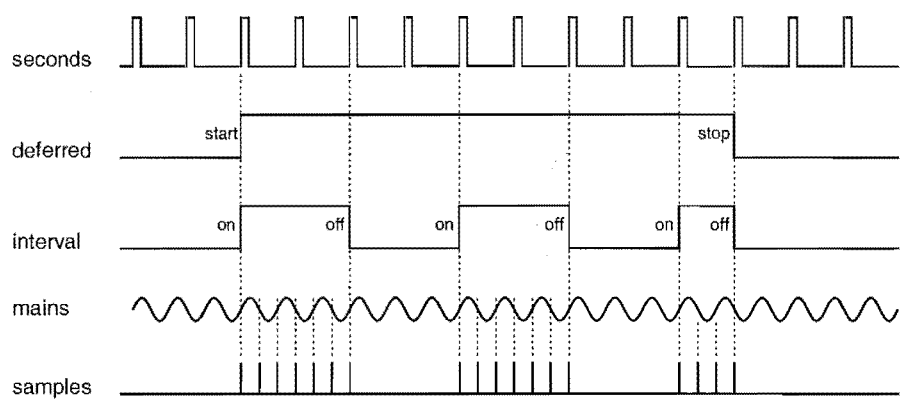
Deferred sampling and interval sampling can be combined. In this case, interval sampling is performed between the start and the stop times given with deferred sampling.

8.2.4 Examples of Using the DSM Sampling Features

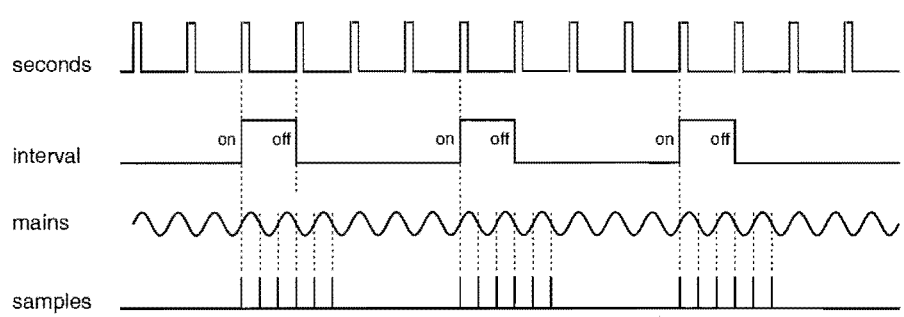
The following example combines burst mode sampling synchronised to the mains, started and stopped at specified times. As long as sampling is started, 4 samples are taken at the beginning of each second. The sampling frequency is a multiple of the mains frequency.



Here deferred and interval sampling are combined. Samples are taken during the on-time of each interval. The on-time of the last interval is shorter because sampling was stopped. The sampling frequency is a multiple of the mains frequency.



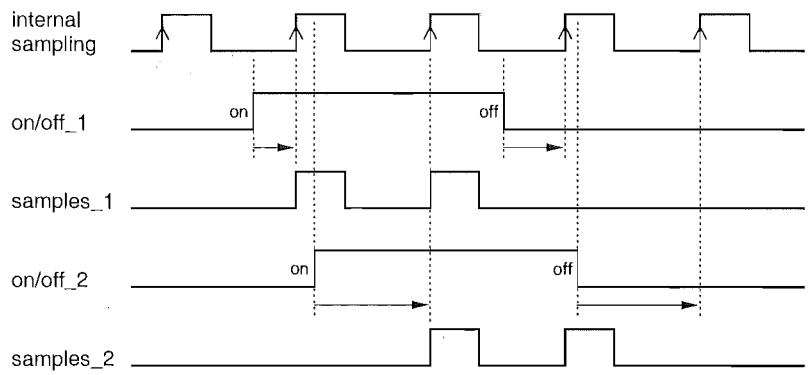
Interval and packet sampling are combined in this example. At the end of the on-time of each interval, the last packet is incomplete; sampling is therefore continued until the specified number of samples are taken. Shown are 3 samples per packet and 2 packets (or 1 packet with 6 samples per packet).



8.2.5 Timing Issues

8.2.5.1 Turning the Sampling On or Off

When sampling is turned on or off, the current sample pulse is always completed. The active edge of the sampling clock is the rising one. In both example 1 and 2 in the timing diagram below, sampling is not enabled or disabled before the next rising clock edge of the internal sampling clock. With the lowest possible sampling frequency of 12207 Hz, the first sample can be delayed up to 81.9 µs. Accurate time stamping is still possible because the vernier always holds the vernier time of the beginning of the current sample.



8.2.5.2 CADU Display Updates

The current DSM application sends the DSM status, system time, and system fundamental frequency to the CADU at the rate of one per second. The CADU program stores the incoming data in a buffer, and displays the buffer contents at the rate of one per second. Because the update of the CADU's display is not event-driven, network delays and a slightly differing clock speed on the CADU can cause the status display to not reflect the true state for the current second. For example, the time can occasionally increment by 2 s, which is balanced by an occasional increment of 0 s.

8.3 FPGA Design

The DSM was designed to be as flexible as possible with respect to the time base functions it provides. For this reason the circuitry of both FPGAs is loaded during runtime, allowing it to change to completely different timing circuits within a few seconds via the DSM setup on the CADU. Two new designs were completed to implement the sampling features described previously. These choices exist for the DSM timing circuitry:

Continuous_old

The same functionality as it was implemented by CHART at the end of 1995. Starting and stopping the sampling requires a GPS receiver to be connected. This FPGA configuration is no longer supported. Its use is discouraged.

Continuous

All timing is based on the internal 1 s vernier counter. Hence, a GPS receiver no longer needs to be connected for a fully operational DSM.

ContBurstPacket

As Continuous, but with added burst and packet sampling capabilities as described in section 8.2.2. This option is recommended because it includes the functionality of the previous one.

8.4 GPS Receiver Operation

A GPS receiver is an integral device for the operation of the DSM. To ease the use of such a complex device, functionality was added to perform a cold start, a warm start, and a re-initialisation via the CADU. To perform these operations, the DSM needs to be setup as master in the DSM setup of the CADU. DSMs not configured as master are assumed to have only the `1pps` and the serial-data-in signals connected, but not the serial-data-out; i.e. such a DSM can receive data from the GPS receiver, but not send any to it. Therefore, a slave DSM will not attempt to send any control data to the GPS receiver.

The following GPS receiver types can be selected:

None

No GPS receiver is connected, or a connected receiver is effectively disabled. No time recovery will be performed with any data received from the GPS receiver, and no data will be sent to the GPS receiver. The `1pps` signal, if present, will still be effective for locking the vernier time. Except for the `1pps` signal, this is equivalent to having no GPS receiver connected to the DSM.

MX4200

Magnavox MX4200. All data communication with the GPS receiver will assume this type of receiver being connected.

The operation of the MX4200 GPS receiver is described in detail in the manuals shipped with the receiver [132, 134], and in the application notes [135, 137–139, 141, 143].

8.5 System Time and Fundamental Frequency

New packet types have been introduced to broadcast the system time and fundamental frequency. Time and frequency are broadcast once per second, independently of the DSM application. Previously data packets had been used for this; now the use of data packets is entirely up to the DSM application software.

In a system with multiple DSMs, only one DSM can output time and frequency information to the system (this is independent from the time stamping). The DSM will only broadcast time and frequency information if it is set up as master in the DSM setup of the CADU. It need not be started, but the frequency information is not available (i.e. 0.0) if the SRM is operating in a way which does not allow determination of the mains frequency.

The fundamental frequency can now be determined when the SRM is operated in either sync-to-mains or free-running mode, as long as the zcross (mains reference) signal is available. When in sync-to-mains mode, the mains frequency is determined out of the setup number of samples per cycle, and an average of the previously determined sampling frequencies. When in free-running mode, the mains frequency is determined out of the setup frequency by counting the number of samples in zcross periods while assuming that the sampling frequency is constant. Because the zcross signal is very noisy, the determined frequency is rather "jumpy". A moving average filter is applied to the zcross signal to smoothe out its jitter. Both SRM modes of operation make use of this moving average filter.

8.6 Control Messages of the Standard DSM Application

A control message is the way in which control information (i.e. commands) are communicated from the operator throughout the CHART system. They can be sent to an application while it is running, to change operating parameters, or to initiate certain actions.

The following control messages have been made available with the standard DSM application:

Enable/Disable sampling

This turns sampling on or off, independently of the parameters given with the setup.

Initialise GPS receiver

Initialise the GPS receiver. This does not perform a reset of any kind, because it is assumed that a reset would take longer than the time which is available to execute a control message ($\ll 1$ s).

The driver software for the MX4200 GPS receiver recognises when the receiver is looking for initialisation after a reset, and initialises the receiver automatically.

GPS receiver warm start

Perform a warm start of the GPS receiver. This typically does not clear any internal settings which are kept over a power-down. No initialisation is performed after a warm start.

GPS receiver cold start

Perform a cold start of the GPS receiver. This typically clears all internal settings which are kept over a power-down. No initialisation is performed after a cold start.

PARIO outputs

Allows control of the output bits of the PARIO interface.

8.7 Front Panel LED

The LEDs on the DSM (and DAPM) front panels are used to indicate the status of the boards [11,12]. The DSM LED operations have been extended, and are now as follows:

off	Error.
toggle 1/s	Normal DSM operation (DSM is started).
off 1/s for 40 ms	DSM operational but stopped, or initial setup not yet received.
flashing > 1/s	DSM is reset. While the DSM software is loaded there will also be some fast flashes.
flashes/pause	The LED flashes on several times, then a pause follows. That is then repeated indefinitely. This is used to indicate an internal DSM error. This error is non-recoverable and the DSM software must be reloaded. The number of flashes is the error number. The following error numbers have been implemented: 2 CPU overloaded 3 MIX initialisation failure 4 Repeated MIX failure to send a packet 5 Completion of DSM test program (for production) 6 FPGA configuration failure (either FPGA) 7 XC3164 FPGA configuration failure 8 Illegal FPGA configuration requested 9 Illegal GPS receiver type requested 10 Serial output transmission failure

8.8 Operating and Application Layer

The DSM now uses a similar structure for its software as the DAPMs [29], with a division between operating layer and application layer. The application layer is a set of functions containing the user program. The operating layer contains the main program, internal functions, and functions which can be called by the application layer. The operating layer can be viewed as a set of library routines which can be called by the application. Every application consists of a set of routines which are called by the operating layer at certain times, like initialisation, or interrupts. The application must not contain a main program (function main in C). A number of functions which every application can make use of has been supplied with the operating layer.

The DSM application currently shipped with CHART resembles the previous behaviour of the DSM, plus the improvements described in this document. Many characteristics described in here are under the control of the application; however, this is not specifically mentioned.

Consult the CHART manuals [11, 12] for further information.

Any DSM application must provide the following functions, and then be linked with the object libraries provided to give the complete DSP program. The variable PASSFAIL can be either PASS or FAIL, as defined in the appropriate C header file.

void App_Init (void)

Initialisation of the application.

PASSFAIL App_Setup (void *setup)

A setup has been sent to the application, setup pointing to the structure. The application should perform type and range checking, and return the result of the processing of the new setup values.

PASSFAIL App_Start (void)

The DSM has been started.

PASSFAIL App_Stop (void)

The DSM has been stopped.

PASSFAIL App_Control (int controltype, CONTROLSTRUCT *controldetails)

A new control value has been sent to the application, controldetails pointing to the control data. The application should perform type and range checking, and return success or failure. The types are defined in the appropriate C header files.

void App_Service (void)

This function is called as often as possible. The application can do whatever is wanted in here, but the processing time required to run through this function once must be far less than one second.

void App_Service_1s (void)

This function is called exactly once per second. The application can do whatever is wanted in here, but the processing time required to run through this function once must be far less than one second.

8.8.1 Status Display

CHART is supplied with a standard DSM application, which is described in its user documentation [11, 12]. A new feature was implemented, that the DSM status is always output by the DSM application as soon as the software is initialised, even before the first setup is sent. The status is also output while the DSM is stopped, but the time and the mains frequency are not output.

8.9 CADU Software Updates

This section introduces two new display functions for the CADU. Both are needed in order to obtain any feedback about the current operational state of the system, and therefore to make efficient use of the features described above. They were also invaluable for developing DSM software. The adoption of the CADU software (running under MS-Windows 3.1) was accomplished by Stephen Hunt, another member of the CHART III development team.

The CADU software is organised as a generic framework, into which functions specific to the DAPM or DSM are loaded dynamically as a library. These specific functions include the display of the data generated by software running on the DSPs.

8.9.1 CADU Date and Time Display (System Clock)

A new display library, called system clock, has been introduced. It performs the same task as the system time display library, but displays the date as well as the time.

8.9.2 CADU Display of the DSM Status (DSM State)

These internal variables of the DSM can be displayed on the CADU by opening the "DSM status display" window:

DSM on	→ YES
DSM master	→ NO
Sampling On	→ YES
Zcross detected	→ YES
SRM locked	→ YES
Mains frequency	→ 50.02 Hz
1pps detected	→ YES
RTC locked	→ YES
GPS msg received	→ YES
Satellites visible	→ 11
Satellites tracked	→ 10
PARIO In	→ 1110000
RTC time (Local)	→ Wed 13 Mar 1996 22:00:00
GPS time (GMT)	→ Wed 13 Mar 1996 22:00:00

Or, if for any reason some values are not available (time_t values equal 0, mains frequency is equal to 0.0, or satellite numbers are negative), the respective lines show:

Mains frequency	→ *
Satellites visible	→ *
Satellites tracked	→ *
RTC time (Local)	→ *
GPS time (GMT)	→ *

"DSM on" shows whether the DSM is currently started, "DSM master" whether it is configured as master, "Sampling on" whether sampling is currently on, and "Zcross detected" indicates whether the DSM has detected a zero-crossing signal on the "Ref" front panel input.

"SRM locked" shows whether the SRM is currently locked to the reference (zero-crossing) signal. This will always be "No" without a zero-crossing signal. Because of noise on the reference signal, this flag usually flickers somewhat. The SRM is deemed locked when the difference between the actual and the expected zero-crossing is below a certain threshold.

The mains frequency displayed is the one of the signal connected to the DSM "Ref" input; no frequency will be displayed without this signal. If the SRM is operated in free-running mode, the

frequency display can be much more unstable than when the SRM is operated in synchronised-to-mains mode.

"1pps detected" and "GPS msg received" indicate whether the DSM receives a 1pps signal, and whether any message has been received over the serial interface over the past second. "Satellites visible" and "Satellites tracked" are the number of satellites visible to the GPS receiver, and how many of those are tracked.

"RTC locked" means that the DSM vernier is synchronised to GPS time. The vernier is used for time stamping with a resolution of less than 1 s (time stamping with a resolution of 1 s does not require a vernier). Synchronising the vernier requires a 1pps signal. As long as the 1pps signal is available, the accuracy of the time stamping is always the best possible, which is approx. 1 µs. Not using the vernier does not reduce the time stamping accuracy—only the loss of 1pps does. The RTC is deemed locked when the actual number of vernier clocks differs less than a set threshold from the expected value, with 1pps used as reference. The threshold is currently set to approx. ±300 ns. Refer to section 8.10 for the implications on synchronous measurements.

"RTC time" is the time as maintained by the DSM, and used on the time stamping bus (TSB) and broadcast to the rest of the system. "GPS time" is the time as received from the GPS receiver. If the GPS receiver is connected, operating, the receiver type is selected correctly, and "use GPS" is enabled in the setup, the GPS time will be transferred to the DSM's RTC time. GPS time is always displayed in GMT (Greenwich Mean Time, i.e. time zone 0). RTC time is displayed as local time.

"PARIO In" is a binary representation of the 3 event inputs, and 4 parallel inputs. The bit order is:

event inputs bits			parallel inputs bits			
2	1	0	3	2	1	0
resulting word						
6	5	4	3	2	1	0

The event inputs represent the actual logic level; if the input stage is not supplied with power the result is "1". The parallel inputs are "1" when the input LED is powered up.

8.10 Synchronised Measurements with CHART

This gives a brief description of a possible setup for synchronous measurements. Application note 001 [106] contains a more detailed example of setting up two CHART systems for synchronised measurements. This application note is reproduced in appendix A. In the following it is assumed that the signal sources are too far apart for using one CHART system for both signals.

Two separate CHART systems are required. They both need their own GPS satellite receiver attached. For a reliable and accurate time stamp the RTC of both DSMs needs to be locked, see section 8.9.2 for details. The coarse time (date, hours, minutes, seconds) also needs to be equal on both units, which is ensured when a valid time has been received during the previous second ("GPS time"), and the use of the GPS receiver is enabled with the setup. Receiving a time from the GPS receiver itself requires the receiver type to be correctly selected.

If it is desired that the start of the sampling of both CHART units is synchronised as well as the time stamping, sampling control must be set to deferred sampling as explained in section 8.2.3. The SRM mode and the sampling mode have no effect on either time stamping or start sampling accuracy. The start sampling accuracy is identical to the general accuracy of the time base.

If the RTC of any CHART system involved in synchronous measurements is not locked, the time difference between the RTC times must be added to the time stamping error (and start sampling error) of all sampled data.

8.11 Conclusion

This chapter has described a number of improvements to the DSM. Several new options for starting and stopping sampling have been added, and allow the sampling time, duration, and number of repetitions to be specified. Sample time stamping can also be performed in the absence of the signal from the GPS receiver, although the accuracy will be reduced to that of the DSM's crystal oscillator. The GPS receiver can be restarted or re-initialised via the CADU.

To offer a convenient indication of possible errors, the LED on the DSM front panel was programmed to display error codes by flashing in specific sequences. Display of all sampling and DSM status variables on the CADU was also implemented.

For consistency with the other DSPs and to make use of the operating / application layer distinction, the software of the DSM was restructured to match the structure of the DAPM software. The application layer contains the user program, and the operating layer contains the main program, internal functions, and functions which can be called by the application layer.

These changes improve the usability of the DSM, increase its functionality, and make it easier to extend functionality further.

chapter 9

Field Testing and Resulting Improvements

The CHART instrumentation system has been used in a number of situations to assist power companies with investigating aspects of their networks. This chapter describes two such deployments with which the author was directly involved, which are typical examples of applications of this system. The focus when undertaking these measurements was on the experience obtained from the process, rather than the data resulting from the measurements.

The first application (section 9.1) tracked the ripple control signal which is injected at three different locations into the city of Christchurch's power supply network. The experimental setup for this application is described in some detail, to demonstrate how such a power instrumentation system is operated, and to highlight the significance of usability and reliability and the need for remote system access.

The second application monitored the harmonic disturbances in a university department and the effect of an active compensation filter over a period of time. This was used to establish limits of the CHART system when used for sampling in the time domain.

A number of issues were observed, especially with the set-up of synchronised measurements. The instrument did not always function correctly during the field tests, as is expected when testing a prototype system. When the system malfunctioned, further iterations of the tests were performed to identify the cause, therefore these malfunctions are described as part of the measurements, in section 9.1.3 for the ripple tests and section 9.2 for the measurements of harmonic disturbances. Other observations from the ripple field tests are discussed in section 9.1.4.

As a result of these field tests, new programs were written for setting up multi-channel synchronised measurements (section 9.3), analysing the resulting data (section 9.4), and capturing data in the time domain (section 9.2). A number of other improvements and fixes to the CHART software are also discussed (section 9.2.1). Finally, to facilitate management and minimise the number of site visits required during sampling, remote access to the CHART system via the internet was implemented (section 9.5).

9.1 Ripple Injection Measurements

9.1.1 Introduction

To investigate CHART's effectiveness in measuring the output current and synchronisation of the ripple injection plants in Christchurch, CHART units were installed at two of the three injection

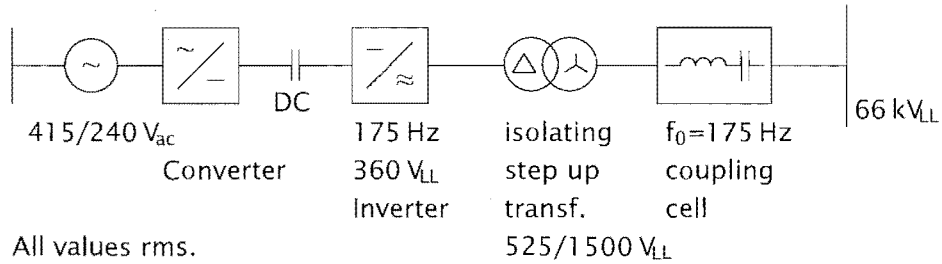


Figure 9.1: Injection of the ripple signal into the distribution system. Also see photographs on page 257ff.

plants. To investigate the current split between the Southpower and Transpower networks, a third CHART unit was installed in the Papanui Transpower control room. Synchronous measurements can establish the synchronisation between two of the ripple plants, and data from all three systems can provide an estimation of the ripple current distribution in the network.

A ripple control signal through the electricity supply in Christchurch controls various functions like street lighting, peak load control, meter tariff changes, and heating control in some public buildings. For example, during periods of high load, water heating can be temporarily turned off. This ripple signal is synchronously injected at 3 points. Southpower, the local electricity supply company, was interested in establishing the output current of the injection plants, and the current split between the Southpower and Transpower networks. The Transpower network is the connection between the power generation plants and the city's distribution.

This ripple signal is a 175 Hz signal superimposed onto the $f_{nominal}$ = 50 Hz power distribution system. It consists of a sequence of pulses of various length, resembling a bit pattern. Further information about this signal can for example be found in [57].

A simplified circuit diagram of the equipment used to generate the ripple signal is shown in figure 9.1. There are three injection plants in Christchurch (Papanui, Pages Rd, Halswell) which are synchronised to a master controller.

Over time, changes to the network, particularly those made to meet increased demands, have lowered the network impedance. This would increase the injection current, thus perhaps overloading the coupling cell. The possibility of a low impedance at 175 Hz somewhere on the network exists. If the injection plants are not correctly synchronised, the ripple energy would be drained by the other injection plants, because each of these plants has a very low impedance at 175 Hz. This also increases the injection current.

Southpower contracted a consultant to determine the output current of each of the three injection plants and the phase differences between the plants, as well as verify modelling results obtained earlier by the University of Canterbury. While this was happening, Southpower also made the installation of two CHART units at the Papanui and Pages Rd substations possible, monitoring the magnitude, and later the phase, of the 175 Hz ripple control signal. Transpower made the installation of another CHART system in their control room possible, so that measurements could be repeated with all three systems sampling synchronously.

Some photographs of the equipment used are presented in appendix H, starting on page 255. The first photographs show CHART installed in the substations, and photographs 13 and 14 show the ripple injection gear.

9.1.2 Instrument Setup

One six-channel system was set up at each of the Papanui and Pages Rd substations. The capabilities of both system configurations were the same. The measurements were later extended, and an additional twelve-channel system was set up in the Papanui Transpower control room, to measure the ripple signal distribution over the Transpower/Southpower networks.

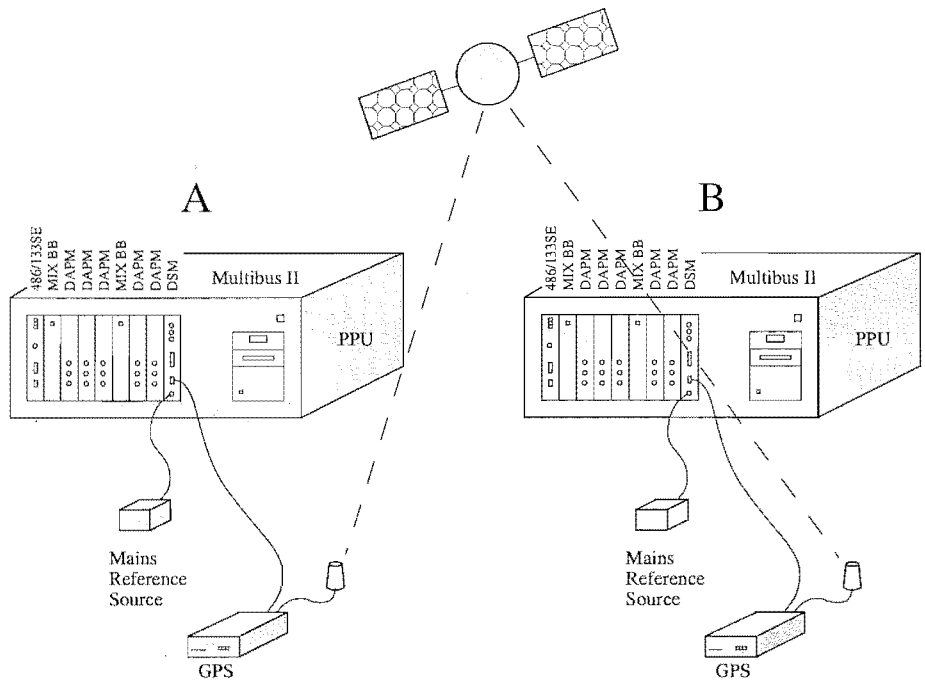


Figure 9.2: Synchronisation of two separate CHART units.

The six-channel CHART hardware configuration at each location included 6 RDCMs attached to 2 DAPMs, a DSM, and a GPS satellite receiver with antenna, providing synchronisation as depicted in figure 9.2.

At both locations, voltages and currents of the red, yellow, and blue phases were measured of the respective 66 kV system. Points in the cabinet at each location represent the 66 kV buses through voltage and current transformers. The RDCMs were connected to these respective cabinet points.

The injection equipment at both locations is housed in a simple purpose-built building with brick walls, no windows, and a flat re-inforced concrete slab roof.

For the first day of measurements at Papanui, the GPS antenna was put up on the flat roof of the building. The GPS receiver had no problems tracking satellites. Afterwards the antenna was kept inside the building, which did not seem to affect GPS operation. When moving the antenna inside, the number of visible/tracked satellites did not change. While a ripple control signal was being injected, the receiver did not noticeably malfunction.

At Pages Rd, the GPS receiver was found to operate with the antenna located inside the building. At times only a comparatively small number of satellites were tracked by the receiver, but this could not be clearly attributed to the antenna being inside and must have been caused by other factors. Moving the antenna temporarily outside did not change the number of visible/tracked satellites. As at Papanui, while a ripple control signal was being injected, the receiver did not noticeably malfunction.

Towards the end of the course of the measurements, a 12-channel CHART system was set up at the Papanui Transpower control room. The hardware configuration included 11 RDCMs attached to 4 DAPMs, a DSM, and a GPS satellite receiver with antenna. The 12th channel could not be used because an insufficient number of fibre-optic cables were available. The GPS antenna was mounted outside, at the entrance to the building. After initial problems with tracking enough satellites for time recovery, the receiver then supplied the DSM with GPS time.

9.1.3 Measurements of the Ripple Control Signal

The first set of measurements was intended to capture the magnitude of the 175 Hz ripple control signal at both the Papanui and Pages Rd substations. The CHART systems at each substation were set up identically as detailed below.

All DAPMS (2 at each location):

Application	→ Ripple magn. meas. V1.0 (18/10/96)

Both DSMS (1 at each location):

Application	→ ContBurstPacket V1.6 (03/96)
Deferred sampling	→ yes
Start sampling time	→ 17 Oct 1996 19:00:00 NZDT (+13)
Stop sampling time	→ 18 Oct 1996 16:00:00 NZDT (+13)
Interval length	→ 0 s
Interval on time	→ 0 s
Sampling mode	→ Continuous
SRM mode	→ Sync to mains
Samples per mains cycle	→ 1024
Sampling frequency	→ 50 kHz
Nominal mains frequency	→ 50 Hz
Number of samples	→ 1024
Use GPS (if possible)	→ yes
GPS receiver type	→ MX4200
Vernier bytes	→ 0
FPGA configuration	→ ContBurstPacket
Master	→ yes
RTC time	→ not set
Re-initialise DSM	→ not set

This first version of the ripple signal analysis software did not include the capturing of the phases. Therefore, only the magnitudes were recorded.

There were no problems at the Pages Rd substation. At the Papanui substation, the GPS receiver malfunctioned and the whole unit recorded nothing. The DSM software could certainly have coped with a GPS receiver going offline (that is partly why the improvements described in chapter 8 were made), therefore the exact cause of the problem could not be established.

When both CHART systems were restarted, an instance of the ripple control was recorded successfully by both systems.

Both CHART systems were configured again as listed above, and set to record for a period of 2 days.

Again, there were no problems at the Pages Rd substation. At the Papanui substation, the GPS receiver malfunctioned again, but data was recorded. In every case, all GPS receivers were functioning correctly when the systems were set up. Unfortunately, because of the malfunctioning time source, the accuracy of the time stamping can only be assumed to have happened with the accuracy of the DSM's internal oscillator.

At this stage, new DAPM software became available which allowed the recovery of the harmonic phase information as well as the magnitude. The configuration of hardware and software was otherwise as before. The duration of the measurement was intended to be 7 days. Because of the long sampling time, additional care was needed to separate the total amount of data into a large enough number of files in order to not exceed the iRMX file size limit of somewhat over 200 MB. Even without a file size limit, it is undesirable to create files several hundred MB in size, especially at a point in time when the average computer was an i486-class machine.

The GPS unit at Papanui malfunctioned again and was replaced. At each location, one DAPM channel had lost synchronisation with the other channels. The situation at both substations was unsatisfactory, and it was decided to restart both systems. The data files at Pages were later found

to have zero length, therefore that unit was restarted again. The remainder of the data recording progressed satisfactorily.

In the next set of measurements, a 12-channel CHART system was installed in the Transpower substation at Papanui to establish the current split of the ripple injection current between the Transpower and Southpower networks. Because channels are limited, to monitor as many buses as possible, only the red phase current of each bus was monitored. The instrument setup at the Southpower substations in Pages Rd and Papanui was unchanged.

With the exception of one faulty DAPM channel in the Transpower instrument, data recording functioned without problems.

Finally, to test whether the apparently-random malfunctioning of CHART instruments at various times was connected with a less-than-ideal power supply, a UPS (uninterruptible power supply) unit was fitted to the two CHART systems connected at the ripple injection plants. Not taking into account unreliable equipment (the first GPS unit at Papanui), the use of UPSs solved almost all of the problems with the CHART gear. Because the mains frequency reference of both CHART systems was accidentally connected to the output of the UPS instead of to the mains supply, the data from these measurements was sampled at the wrong frequency. The use of the UPS still demonstrated the importance of a reliable power supply and/or UPS for such instruments.

9.1.4 Results and Discussion

A total of 1.8 GB of data was recorded during the field tests. An initial analysis of this data was performed with the new chartdat program (which is described in section 9.4). Results showed the principal usefulness of CHART for this type of measurement, and highlighted possible improvements in the area of usability, reliability and data analysis.

These field tests showed that the use of a UPS for each instrumentation system is mandatory if there are any doubts about the quality of the mains power supply. However, the malfunctioning of CHART at various times could not be completely explained by problems with the mains power supply. The exact cause could only be determined by more measurements in the substation itself, for example investigation of the influence of strong magnetic fields in the vicinity. This malfunctioning never occurred in the laboratory, and has not been observed during other field measurements with CHART. A repeat of an identical system configuration was performed in the laboratory and did not show any signs of malfunctioning.

The process of setting up the sampling with the CADU was found to take up a substantial amount of time when multiple channels were involved. The CADU software was programmed generically to handle setting up every possible DAPM analysis software, in anticipation of possible future applications. To achieve this, the type and other associated values of every DAPM setup parameter was queried by the CADU for each DAPM channel. Because the CHART software was initially completed only to a stage where the principle functionality of the system could be demonstrated, few optimisations for speed or to increase practical usability had yet been undertaken.

A large number of mouse clicks are required to set up one channel with the CADU. There are no facilities for operating on more than one channel at a time. It is desirable to double-check the configuration of each channel before the commencement of measurements, especially when multiple locations are involved, but there is no feature to list the current setup status of one or more channels. Therefore checking requires an equal number of clicks to setting up. Instabilities in the operating system underlying the CADU are an additional source of inefficiency.

While it is possible to quickly record a period of data to the same previously configured file names by manipulating the DSM settings for enabling the generation of the sampling clock, this does not re-synchronise the DAPMS. In particular, it does not cause the last data package of the previous recording interval to be filled and correctly time stamped, and it does not cause the new recording interval to store its data starting with a new packet. To guarantee correct synchronisation, all DAPMS need to be restarted.

A new program was written to address some of these issues, and is described in section 9.3.

Number Channels	Packetsize (128 points/cycle)	Number Cycles	Number Cycles ok	Data Volume	Notes
1	5	3000	3000	1.5 MB	
3	5	3000	3000	4.5 MB	[a]
6	5	3000	1010/1100	3 MB	[a]
12	5	750	250	1.6 MB	[c,b]
12	5	250	250	1.5 MB	[c]
12	5	300	≤ 200	1.4 MB	[c,b]
1	8	300	$\approx 60 \cdot 8$		[d]
1	7	300	$\approx 70 \cdot 7$		[d]
1	6	300	$\approx 84 \cdot 6$		[d]

[a] All channels on a single MIX baseboard.
[b] System stopped with protection fault.
[c] A further 6 channels are on a MIX baseboard with DSM.
[d] Data distorted, i.e. not recorded correctly

Table 9.1: Time domain recording limits, given as number of mains cycles. Only the DSPs in use were loaded with software.

The repeated issues which required multiple site visits to investigate and rectify demonstrated the usefulness of connecting a power instrumentation system to the internet for remote management. This was undertaken for CHART and is described in section 9.5.

9.2 ELEC Feeder Measurements

The electricity for the Department of Electrical Engineering at the University of Canterbury is supplied from the university's ring feeder. Because of the large number of computers and their switch mode power supplies, the peaks of the supply's waveform are noticeably flattened. This provided a good testing environment for an active harmonic compensator which was developed by a local company.

During the compensator tests, a CHART system was connected to the feeder line, measuring voltages and currents of all three phases. Of interest to the developers of the compensator was the static behaviour of the active filter over a period of at least a week, its effect under the different conditions found in the department, and its transient characteristics. For the scope of this thesis, the undertaking provided an opportunity to apply CHART to collection of data in the time domain. This illustrated requirements for such measurements and resulted in the establishment of CHART's limits for time domain recording.

To record transients, time domain data needs to be stored instead of the harmonic data. At that point, there was no DAPM application for this. The existing DAPM application for harmonic analysis, harmac, was modified appropriately and called contime. The 8-to-1 FIR decimation filter was left unchanged. Data is output by contime at a rate of 128 points/cycle, 10 cycles/packet, and 5 packets/s.

Because the vastly increased data rates of recording time domain data noticeably exceed the continuous recording limit of the PPU, recording time domain data only works for a certain length of time. This length of time is heavily dependent on the number of channels involved. Essentially, it runs until all buffers are filled up. To maximise the limit, the HUB processor board of the PPU was equipped with the maximum amount of memory possible.

The GPS receiver did not work inside the building housing the utility. Because only a single instrument was involved, highly accurate time stamping was not required. Sampling coherency between channels of the same PPU is implicit in the design. No malfunctioning of any equipment was observed.

Number Channels	Packetsize (128 points/cycle)	Number Cycles	Number Cycles ok	Data Volume	Notes
3	10	500	500	0.8 MB	
6	10	500	500	1.5 MB	
12	10	500	500	3 MB	
12	10	1000	630	5.4 MB	
12	10	620	620	3.8 MB	
12	10	650	650	4 MB	
12	10	700	630	4.3 MB	

Table 9.2: Time domain recording limits, given as number of mains cycles. This is after the implementation of asynchronous packet transfers.

When examining the limits of recording time domain data, the values shown in table 9.1 were obtained. A 6-channel system could record up to 1000 mains cycles, and a 12 channel system perhaps around 100. A reliable upper limit proved too difficult to obtain. When examining the recorded data, the sample signal was not recorded correctly. This turned out to be caused by ring buffer overruns in the DAPM's operating layer software. Possible solutions were identified as:

1. Reduce sampling rate to 512 points/cycle. However, the butterworth filters in the DCMS only support 1024 points/cycle.
2. Reduce packet size to 5 cycles $\times$ 128 points per packet.
3. Modify the DAPM operating layer software to make the data package function call asynchronous. Implementation would require a periodic interrupt, transferring data if the transfer queue were not empty.

   The previous implementation assembled the data packet, and then looped until the last byte was transferred. This took too long, causing the buffer to be filled again before it was fully transferred.

Because it is questionable whether the first two solutions could make the system operate reliably, the third solution was implemented, allowing the limits stated in table 9.2. With 12 channels, a minimum of 10 seconds (500 cycles of the mains) of time-domain data can be recorded reliably. It appears that the limits are mostly independent of the number of channels, indicating that the limiting factor is closer to the DSPs than the HUB.

9.2.1 Other Improvements to the CHART Software

A number of other improvements were made to the CHART software in the course of the active filter measurements (section 9.2).

A new gcc-based compiler for the Texas Instruments TMS320C31 DSP became available. Compiling the CHART DSP software with gcc showed a better performance of the front-end's syntactical and semantic analysis of the source code than the TI compiler. For several C constructions which were syntactically correct but almost certainly erroneous, gcc issued a warning whereas TI C gave no indication of a potential problem. These constructions include comparing variables which can only hold positive values for ≥ 0, and shifting a 32 bit variable 10000 times to the right.

The DAPM applications harmac and contime and the DSM application contbp were recompiled with the gcc compiler, and all potential problems were fixed.

All data transfers between CHART CADUS and PPUS are handled by a library called chartnet. This library was programmed in such a way that it was expected to be platform independent, however this had not yet been tested. To increase the number of platforms which can be used to

interact with CHART equipment, the chartnet library was ported to ix86 Linux and SPARC Solaris for use with the getsync program (discussed in section 9.3). The SPARC architecture is big endian, whereas the ix86 architecture is little endian. This provided a good testing environment for the platform-independence of the CHART software, and several minor issues were rectified. Endianness-independence of software can never be relied upon until fully tested. This point was emphasised again when programming the chartdat program (discussed in section 9.4).

From the point of view of efficiency and portability it is also important to design data structures so that they are optimal for different conditions. This affects the alignment of data in structures. 1-byte values are never problematic. On a 32 bit architecture, storing 2-byte values at an even address does not cause problems, but storing them at an odd address causes a misaligned memory access. Likewise, storing a 4-byte value at an address which is not divisible by 4 creates a misaligned access because the 4 bytes can not physically be accessed by a single access. On some architectures, including SPARC, this causes a fatal bus error. On other architectures, including M68000 and ix86, access to the 4-byte value has to be split into two separate accesses, decreasing execution speed.

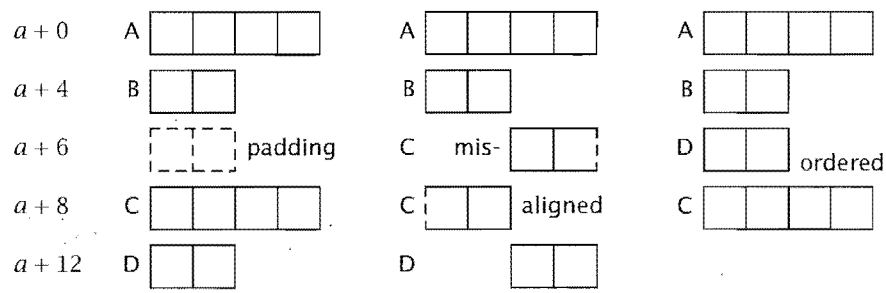


Figure 9.3: Alignment of a data structure with members A to D in memory at address *a*. The left structure was padded with 2 unused bytes between members C and D. In the centre, the padding was suppressed with the `packed` keyword. This program will not run on all architectures. On the right, problems were avoided by swapping members C and D.

To avoid these problems, compilers try whenever possible to align data storage, if necessary by padding structures with unused bytes, as shown in figure 9.3. The order in which structure members are entered into the structure is not irrelevant, and some of the CHART structures are not optimal in this respect.

9.3 The Getsync Program

The getsync program interacts with CHART HUBS and is effectively the equivalent of a CADU. It was programmed to address some of the issues identified in sections 9.1 and 9.2.

The main feature is that multiple channels can be set up simultaneously and quickly (within the limits of the data transfer speed). Full control over the DSM is possible. Getsync is also designed to be portable across computer architectures, and is tested on ix86 Linux and SPARC Solaris, thus offering a stable working environment. It does not currently use a graphical user interface (GUI); this could be added later either by incorporating it into the program, or by implementing the GUI as a separate layer, calling getsync as a helper program. To retain maximum flexibility of application software, it is important that the non-GUI functionality is retained.

Getsync is able to perform all steps necessary to efficiently handle a HUB system for measurements to be undertaken, namely:

- Start/stop/reset processors
- Load application software onto processors

```
Usage: getsync [OPTIONS] PPU [PPU_PORT]
VK  V3.0.3  11 Jun 1998                                    GNU getopt()

OPTIONS:
  -h,--help           show usage

  --hwinfo            show hardware,          all / start at --addr
  --swinfo            show software running,  "
  --dsinfo            show data sets,         "
  --ctrlinfo          show app SW controls,   "
  --setupinfo         show app SW setup,      "
  --storeinfo         show PPU storage setup,"

  -t,--systime        show system time, use DSM --addr if given
  -f,--sysfreq        show system frequency,                  "
  -s,--dsmstatus [N]  show DSM status (N times, default 1), "
  -e,--dsmevent [N]   show DSM events (N times, default 1), "
                      Displaying these DSM data sets is not (yet?) recursive!
  --dump SET          hex dump data set SET of DSP --addr
                        (SET+1000: hex, SET+2000: float, SET+3000: float 1 col)

  -R,--reset          reset DSP --addr
  -L,--load           load --application into DSP --addr
  -U,--update         update DSP --addr
  -S,--start          start DSP --addr
  -T,--stop           stop DSP --addr
  --Load              same as --reset --load --update --start
  -N,--Rename         rename data sets of DSP --addr
  --smpoff, --smpon   turn sampling off/on (is DSM SW control)
  --parput, --parset, --parclr VAL  control DSM PARIO outputs (DSM SW control)
  --numcycles N       sample N cycles in time domain (is DAPM SW control)
  --control NUM[,TYPE,VALUE]  send SW control NUM, with TYPE + VALUE if given
  --param NUM,TYPE,VALUE      send SW setup parameter NUM, TYPE, + VALUE
  --storesetup NAME:STORERATE:DSADDR[:DSADDR...]  store datasets in file
  --storesetup NUM    close storage setup NUM

  -d,--addr HWADDR    use this DSP address, multiple addresses separated by colon
  -a,--all            change --addr to root (i.e. all in PPU)
  -r,--repeat N       repeat operation N times
  --application NAME  name of DSP application software to load
  --DAPM              change --appname, --addr resource default to that for DAPM
  --DSM               (default) as --DAPM but for DSM
  --user NAME         use this name to log into PPU
  --passwd PWD        use this password to log into PPU
  --testloginout      test logging in and out

  HWADDR: x,x,x,x,x,x,x,x (trailing 0 can be omitted)
  DSADDR: HWADDR/N/DATASETNUM (N is ignored and can be left out)
  TYPE: as number (enum value) or string (as printed by --...info)
  VALUE: format depends on type

PPU       PPU name (as known by DNS), or dotted decimal
PPU_PORT  PPU TCP/IP port to use (default 59395)

*** Warning: do not run some operations on some DSPs! ***
*** Some characters might be special to the shell and must be quoted ***

Resource files: getsyncrc, ~/.getsync.
```

Figure 9.4: The help output of getsync (when called with -h).

- Modify setup and control parameters of any processor, for example
  - DSM: sampling frequency, times, duration, etc.
  - DAPM: input scale factor, etc.
- Configure DAPM channels for sampling and data storage (via their setup and control parameters)
- Query any processor for its status, for example
  - DSM: GPS time, mains frequency, etc.
- Control the PARIO outputs of the DSM and display the value of the PARIO inputs
- Perform an operation on all, or on a given subset of processors

In particular the ability of multi-channel operations is not available with the MS-Windows based CADU program.

9.4 The Chartdat Program

A new tool was programmed to analyse huge quantities of data more quickly than the existing postcess utility, and in a way which can be automated. Postcess can display data graphically and save it as text. Chartdat does not display any graphs, but can apply a number of operations to the data, and more importantly to parts of the data, and output the results. Results can then be processed by any graphing or analysis tool of choice.

Chartdat is a command-line oriented tool, which makes it easy to port to other platforms, including those with a faster processor than an ix86. It was originally developed when the average PC was a 100 MHz Pentium with 32 MB of memory, while fast SPARC workstations in the department were running faster and had hundreds of MB of memory. Ports to Linux (for ix86 and SPARC), Solaris on Sparc, and MS-DOS were completed. Ports to other platforms would be easy to develop because the programming is compliant with the ANSI-C standard.

Because chartdat does not require interactive operation, it is suitable for autonomous analysis of the data. The host it is running on can be of any supported type, can be significantly more powerful than either of HUB or CADU, and does not need a display attached (because chartdat can be run remotely, or autonomously). Data recorded by HUBS in different locations can be processed by chartdat if the data can be transferred over a suitable network (for networking see section 9.5).

The operations which chartdat can perform are separated into actions and limits. Exactly one action must be used, and any number of optional limits can be applied to restrict the operation to a subset of the data.

The available actions include: info (display all information about a data file), copy (copy a data file), view (display the contents of a data file, which is also possible with partial information about the data), statistics (display some statistics for each data packet in the file), and fix (fix a data file which was not properly closed, which happens when a CHART unit is shut down without deleting all data sets first).

The statistical output computes some basic statistical values, such as mean and standard deviation. This can be very useful when searching for values in the data which vary considerably from the average.

The available limits include thresholds, times, and data sets. The threshold function outputs the time of every data packet with at least one value exceeding the given threshold. The data set limit restricts the action to a subset of the data sets present in the data file, and the time limit restricts the action to those data packets which meet the time limit. A time range limit is allowed as well

as a time list limit (the time list could, for example, have been created by a previous run with the threshold limit). All these limits can be combined with each other.

Complete instructions on how to operate chartdat can be found in the manual [107], which is reproduced in appendix B. A practical example of the use of the actions and limits of chartdat is given in the next section.

9.4.1 Analysing the Ripple Data With Chartdat — An Example

This section assumes familiarity with the chartdat program. For a detailed description please refer to its manual [107] or appendix B.

When the ripple-injection signal is measured with a 6-channel system, where the 6 channels are connected to voltage and current transformers of a single 3-phase system, magnitude and phase information comprises approximately 50 MB/24 h of data. This shows the importance of being able to analyse sample data efficiently.

The duration of the ripple injection is, very roughly, one minute, and occurs on average once per hour over a 24 h period. The injection times are unknown in advance. With only 1/60 of the data being of interest, copying those data packets containing ripple data over to a new file considerably reduces the amount of data to be handled.

To find the injection periods, chartdat can be used as outlined in this example. The injections show up in the data sets of the voltage and current magnitudes as values significantly higher than the average value created by noise. The "statistics" action of chartdat gives the mean and standard deviation of all values in a data packet. This action should be restricted to the data sets containing magnitudes. Mean and standard deviation of harmonic phase angles are irrelevant. The mean of those packets containing data of an injection should be noticeably higher than for the other packets. The actual value from the injection can be used to determine a suitable threshold.

After a suitable threshold is chosen, this threshold is then used with the "display times" action and the "threshold" limit. The "threshold" limit filters out all packets which have at least one value exceeding the threshold. The "display times" action displays the times of packets, and must be limited to those data sets containing harmonic magnitudes. This reliably finds all packets containing ripple injection data, and possibly a few more depending on the threshold used and the noise levels.

The output, a list of times of those packets containing ripple data, is saved in a file. This file is then used in conjunction with the "copy" action and the "limit times" restriction. This copies all packets which have a time equal to one of the times in the times list file. The "copy" action should not have a data set limit so that it copies the magnitude as well as the phase data sets to a new file.

The result is a file which contains only packets with ripple injection data, and is approximately 1–2 magnitudes smaller. The view action can then be used to output the binary contents of the file to tables of data in text format which can be imported into spreadsheets, graphing or mathematical software.

9.5 Connecting CHART to the Internet and Other Networking Aspects

When measuring in more than one location with the intention of processing the acquired data at a centralised point in real-time, the data must be transferred continuously to the central processing point. Dedicated connections via, for example, microwaves are too expensive and too inflexible. Using the internet is the best solution with respect to access points and setup costs. If there is no internet connection in the direct vicinity of the measuring location, as, for example, in a

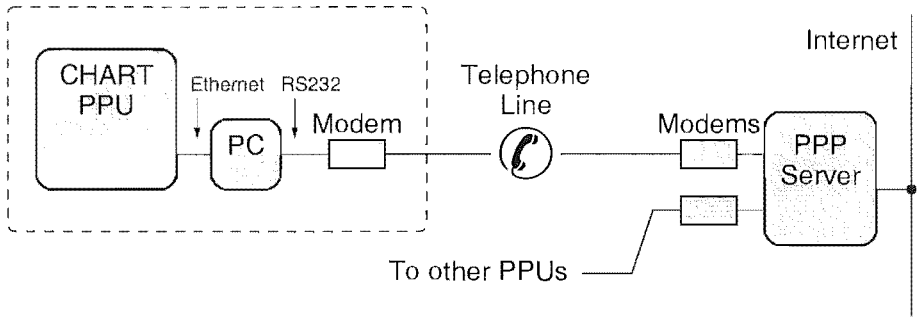


Figure 9.5: Block diagram of an internet connection for a PPU. The boxed items on the left indicate the equipment at each remote measurement location.

substation or switch yard, a serial connection like SLIP (serial line internet protocol) or PPP (point-to-point protocol) can be used to the nearest internet access point. This serial connection can be run via modems over phone lines, which are generally readily available. If Ethernet networking is installed at the measurement location, high data rates are readily available with minimal effort, and slow communication over serial lines is not needed.

The amount of data which can be transferred over the network connection is dependent on the speed of the serial link, and the actual network speed. The speed of the serial line is limited by the modem speed, currently approximately 56000 baud maximum (line dependent), whether data compression is used and what compression ratio can be achieved (data dependent), and the quality of the telephone line (location dependent). A very noisy line will not support as high a baud rate as a higher quality line. With a typical setup where CHART units are placed in substations, the network throughput is mostly limited by the quality of the telephone lines, which tend to be very noisy.

If data security is of concern, data should be encrypted before transmission over the internet, although this will have to be traded off against the total processing power available to each CHART system and to the central processing station.

The central processing is independent of the central display unit (CADU), and may or may not be run on the same system as the CADU. This allows maximum flexibility for the system with respect to utilisation of existing software as well as available hardware. The results can be displayed on the CADU while a fast machine somewhere else on the network does complex processing of all the data acquired at various measurement locations.

For the purpose of using a serial line as the transport medium for an internet-type network, two protocols are readily available: SLIP, and the more modern and more advanced PPP. Both are suitable for CHART, but PPP might be fractionally faster and offers error detection facilities.

The operating system used for the CHART PPU, iRMX, only supports SLIP. Unfortunately, there were problems with the iRMX implementation of this which could not be resolved in conjunction with the iRMX technical support. Other solutions had to be found.

Because software to run a serial line network is not available for CHART, an additional computer is used, which is connected to the PPU via a crossed Ethernet cable and to the internet via a PPP modem connection, as shown in figure 9.5. This additional computer has the advantage of also being able to encrypt the data, allow encrypted control connections to the HUB, and act as firewall. PPP can easily be combined with encryption [22].

Before describing the network setup, the reader is reminded of these terms:

route A path to a particular destination. If a destination is not found in the list of routes, the default route is used instead. A host route is a path to a particular host, a net route is a path to a particular subnet. The default route is usually a host route to the gateway host.

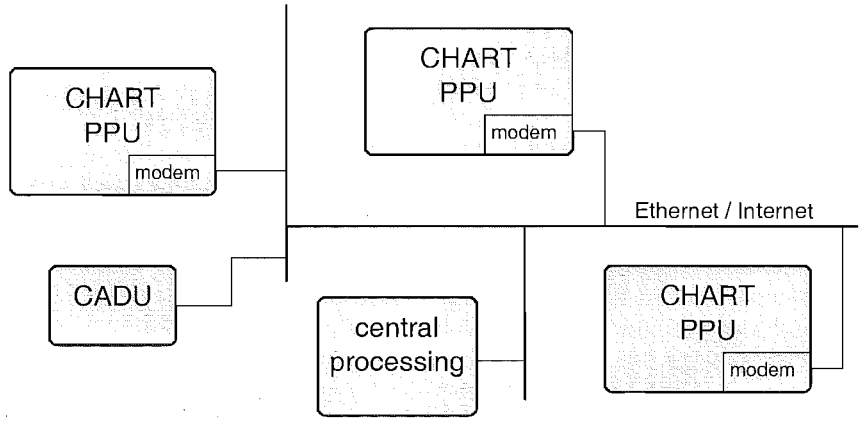


Figure 9.6: Several CHART units networked with a CADU and a central processing host. The CADU and the central processing host can be the same machine, but do not have to be.

subnet A local network connecting a number of machines directly with each other. A gateway is then used to connect this subnet to another subnet. In the dotted notation (xxx.xxx.xxx.xxx), the last number is commonly 0 for a subnet.

gateway A host which connects 2 different subnets together.

ARP Address resolution protocol. Each Ethernet interface has an associated hardware address through which it is addressed. To find out the hardware address, the sending host sends an ARP request which is answered by the addressed machine with its hardware address.

proxy-ARP The same as ARP, but on behalf of another machine which is not physically connected to the local Ethernet network, but via PPP, etc.

The setup of the PC which translates the PPU's Ethernet to PPP (translation PC), and that of the PPP server, are not very complex. The packet routing of all systems involved must be set up respectively adjusted to the network configuration with the PPP link. In addition, the PPP server must service ARP-requests for all the translation PCs as well as all the PPUs connected to this server via these PCs. In detail, the setup is as follows:

On the PPU, a host route points to the translation PC, and the default route points to the PPP-server with the translation PC as gateway. As there are no other systems on this branch of the network, no further routes are necessary. These settings can be put in place automatically after booting. They are not the same as with a direct connection to the internet. Two command files change between the settings: `serialstart.csd` and `serialstop.csd`.

The translation PC has a host route to the PPU, and a default route to the PPP-server. As there are no other systems on this branch of the network, no further routes are necessary. These settings can be put in place automatically after booting.

The PPP-server has a host route to each translation PC and to each PPU served by it. Its default route remains unaffected, and usually points to the gateway connected to the particular subnet the PPP-server is on. In addition to the host routes the PPP-server must answer ARP-requests for all connected translation PCs and PPUs on their behalf. This is necessary so that other hosts send their packets destined for a PPU to the PPP-server, which will then forward them to the translation PC which will forward them to the PPU.

Further information can be found in [38, 51, 90], and a good summary is in [114].

Which hardware and which operating system are used for the PPP-server and the translation PCs is irrelevant, as long as they offer TCP/IP and PPP. All modern operating systems do this.

Although irrelevant, the setup which was developed used the following equipment. The PPP-server and translation PCs are standard IBM-compatible systems running the UNIX-variant Linux. All

this software is freely available and incurs no licensing costs, and the hardware is among the cheapest available. As with all UNIX-systems, the network functions are powerful, and the systems themselves are multitasking as well as multi-user. This means that the translation PCs can be logged into via telnet (or better ssh [207]), allowing remote-maintenance of the system as well as some remote trouble-shooting of the PPU. A watchdog for the Linux system as well as for the PPU could be implemented easily on the Linux system. Linux is a very stable operating system, and the PPP setup is such that the link is automatically re-established in case it fails.

The minimum hardware requirement for the translation PCs is: 80386, 4–8 MB RAM, 100–200 MB disk. For the PPP-router, a fast 80386 or 80486 should be sufficient. It is important that all serial ports are equipped with 16550A UARTs which have a 16-byte FIFO, otherwise characters might get lost at higher serial speeds. Up to 3 modems can be connected easily to an IBM-compatible PC, up to 4 if there is no mouse. After that, special multi-port serial boards have to be used. These boards require special drivers and only occupy one IRQ line for all the ports on the board.

If a cable connection is feasible, the modems could be eliminated. The cable must be a null-modem cable (i.e. have TxD/RxD crossed), and must also have the handshaking lines RTS/CTS (also crossed). This results in 5 wires plus shield for the cable, including GND.

9.6 Conclusions

The two field tests provided experience in actual use of the equipment and lead to the identification of a number of improvements in both functionality and usability. The tests also showed that CHART III can be used for synchronous distributed measurements in the frequency and time domain. The measurements in the time domain established that 10 seconds of data could be recorded reliably with the amount of memory available in the CHART III system.

The ripple injection measurements showed the importance of a reliable power supply, and therefore the need for a UPS when the quality of the supply may be questionable. Another significant observation from these tests was that, for applications requiring many channels, major gains in usability could be achieved by producing software which could control multiple DSMs and DAPMs at once. The program getsync was created to achieve this. Speed of data processing was improved by the chartdat program, which allows data filtering and automated analysis.

If the equipment malfunctioned it could only be examined and restarted on-site, which was sometimes inconvenient. This demonstrated the usefulness of remote configuration over the internet, which was also implemented.

All the issues resulting from these field tests have been incorporated into the discussions on system requirements in chapter 4 and the checklist template in chapter 5.

chapter 10

Simulation of Quantisation Errors for Harmonic Analysis

10.1 Introduction

Designers of instrumentation systems are interested in the extent to which each part of the system contributes to the final error. Harmonic analysers report the magnitude and phase for each harmonic order, therefore the error for these instruments is the differences in both magnitude and phase between the values reported by the instrument and the actual values being measured. A typical structure of a harmonic analyser is shown in figure 10.1. The final error is mainly made up of contributions from:

- ADC (analog-to-digital converter) quantisation
- ADC non-linearities
- Analog electronics non-linearities

and is also influenced by:

- Transform length (the length of the FFT, fast Fourier transform)
- Oversampling ratio
- Number of sample points (which is equal to transform length times oversampling ratio)
- Digital filter (downsampling filter) characteristics

This chapter describes simulations of the errors inherent in the ADC quantisation with respect to their effect on harmonic analysis. The results provide designers of such instruments with information about the hardware required for a desired level of accuracy, and allow customers to assess the significance of results. Average and maximum errors are of interest for both magnitudes and phases. A paper has been compiled about these simulations, and can be found in appendix A on page 170.

The data acquisition stage (ADC) and fast Fourier transform (FFT) performance are simulated with particular emphasis on phase recovery. Phase information is important for establishing harmonic power flows. The simulation variables are sample rate (samples per fundamental period) or FFT length, ADC width (bits), the ratio of harmonic to fundamental amplitudes, and the harmonic

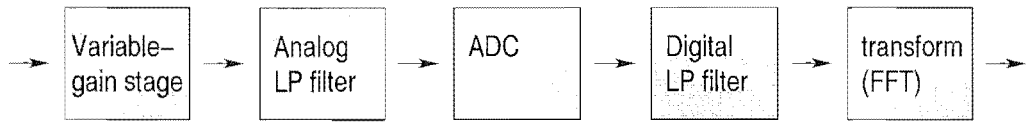


Figure 10.1: Components of a harmonic analyser. The digital low-pass filter is present if oversampling is used.

order. Realistic known input values are fed into the simulation model, and the errors are calculated as the differences between the input and the output values. An ideal ADC is assumed for the simulations; other distortions resulting from non-linearities, sample and hold windows, etc. are not considered.

Measuring current harmonics on power distribution lines presents the problem of a largely varying signal amplitude. For line currents it ranges from near 0 (no load) to typical full load levels, whereas the voltage signal is practically constant. Fault levels are higher for both currents and voltages. Although in the case of a fault the harmonic content is not of primary interest, capturing the fault waveform is of interest for fault location applications. If transient capturing is not part of the objective of the instrument, the higher amplitudes of transients need not be considered. Automatic gain adjustment can be used to respond to changes in input signal amplitude, but has a number of disadvantages, which are discussed in section 4.3. The implications of a system without such adjustment are investigated as part of the error simulations in this chapter.

Resolution issues in detecting the harmonics are a further potential problem, because the amplitude of the harmonics may be orders of magnitude lower than the amplitude of the fundamental. Filtering out the fundamental causes the relationship of harmonic amplitudes to fundamental amplitude to be lost, and requires sharp filters which are expensive to implement with analog components. The simulations presented here investigate to what extent the presence of a large fundamental affects the recovery of harmonic amplitudes and phases.

10.2 Theoretical Considerations

Quantisation by the ADC adds error to the sampled values. If the input signal is not correlated with the quantisation levels of the ADC, the quantisation error is effectively random and uncorrelated, and behaves like additive noise. The input signal is uncorrelated with the quantisation levels if it is spectrally dense, that is, if a number of frequencies in the signal have amplitudes larger than the quantisation interval of the ADC. Also, any signal can be made to be uncorrelated to the quantisation levels by dithering. Dithering is adding a small amount of random noise to the signal before conversion. The amount to add is roughly comparable to the quantisation error [226].

If the ADC has an ideal transfer function as shown in figure 10.5, with a quantisation width of w, the quantisation error is uniformly distributed in the interval $[-\frac{w}{2}; \frac{w}{2}]$, and has zero mean and variance $\frac{w^2}{12}$. The quantisation noise power q is equal to the variance of the quantisation error

$$q = \frac{w^2}{12} \tag{10.1}$$

Since the quantisation errors in each signal sample are effectively random, the quantisation noise in the spectral domain is complex Gaussian, i.e. its real and imaginary parts are each Gaussian distributed with zero mean, and are statistically independent of each other ([24] section 9.2 p. 316). This is so because each spectral output of the FFT is a weighted sum of the signal samples input to the FFT, so the central limit theorem applies to the FFT outputs ([19] p. 325). The central limit theorem states that the distribution of the means of sets of samples from a population of independent variates approaches a normal distribution, independent of the distribution of any one sample set [170].

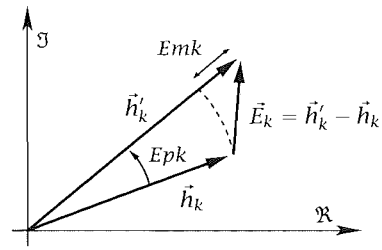


Figure 10.2: The error vector of the quantisation, $\vec{E}_k$, as difference between the measured vector $\vec{h}'_k$ of magnitude and phase, and the actual vector $\vec{h}_k$. Em_k is the difference of measured and actual harmonic magnitudes $Em_k = |\vec{h}'_k| - |\vec{h}_k|$, and Ep_k is the difference of harmonic phases $Ep_k = \angle\vec{h}'_k - \angle\vec{h}_k$. k is the harmonic order.

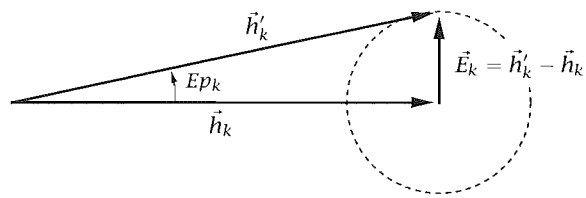


Figure 10.3: The trigonometry underlying the phase error Ep_k, to demonstrate the relationship of phase error variance to harmonic magnitude $|\vec{h}_k|$.

Since the quantisation errors in each signal sample are uncorrelated with each other, the quantisation noise in the spectral domain is white, i.e. the quantisation noise power q is spread equally across all frequencies, including DC[1]. For an N-point FFT, the noise power in each output "bin" of the FFT is $\frac{q}{N}$. The FFT output bins represent the magnitudes and phases of a complex-exponential decomposition of the input signal. The nth harmonic in the input signal appears in both positive-frequency and negative-frequency bins in the FFT output, with the amplitude in each bin equal to half the amplitude of the harmonic, as described by the trigonometric identity

$$\cos x = \frac{1}{2}(e^{jx} + e^{-jx}) \tag{10.2}$$

Therefore, when using an FFT for harmonic analysis, harmonic amplitudes are measured by taking twice the value of the positive frequency bins of the FFT output, and the noise power or variance for each harmonic except DC is

$$Q = \frac{4q}{N} \tag{10.3}$$

Each harmonic component of the quantised signal can be described as a vector with an associated length (its magnitude) and angle (its phase). The diagram in figure 10.2 shows in cartesian form the true value $\vec{h}_k$, the measured value $\vec{h}'_k$ arising from the quantised signal, and the measurement error

$$\vec{E}_k = \vec{h}'_k - \vec{h}_k \tag{10.4}$$

for each harmonic order k. The measurement results a harmonic analyser reports to its user are the magnitude and phase of $\vec{h}'_k$.

The statistics of the noise in the measured harmonics are those of the error vector $\vec{E}_k$, which as previously noted has a complex Gaussian distribution. The statistical properties of $\vec{E}_k$ are the same for all k, except $k = 0$ (DC). Its magnitude has a Rayleigh PDF (probability density function) with variance Q, and its phase has a uniform PDF over the whole circle ([24] section 9.2 p. 316). The measured harmonic value $\vec{h}'_k = \vec{h}_k + \vec{E}_k$ is complex Gaussian, plus a constant vector. Its magnitude has a Rician PDF ([24] section 14.3 pp. 533–534).

[1]Parcival's theorem states that the total noise power of the FFT input equals that of the FFT output. The FFT output is a sum weighted $\frac{1}{N}$ for each spectral bin, therefore the error power (variance) Em_k is inversely proportional to the FFT length. Intuitively, for white noise, a higher number of bins take up less distance each.

Of interest to the user are the magnitude error

$$Em_k = |\vec{h}'_k| - |\vec{h}_k| \tag{10.5}$$

and phase error

$$Ep_k = \angle\vec{h}'_k - \angle\vec{h}_k \tag{10.6}$$

The magnitude error Em_k has a distribution of Rician form, but with the origin shifted by $|\vec{h}_k|$. For large $|\vec{h}_k|$ (harmonic power $\gg$ noise power, $|\vec{h}_k| \gg |\vec{E}_k|$) the distribution of Em_k becomes approximately Gaussian with a mean of zero and a variance of $\frac{Q}{2}$. For small $|\vec{h}_k|$ (not much larger than the noise power), the form of the distribution becomes skewed and the mean significantly larger than zero.

The trigonometry underlying the phase error Ep_k is shown in figure 10.3. In the right-angled triangle shown,

$$\sin(Ep_k) = \frac{|\vec{E}_k|}{|\vec{h}'_k|} \tag{10.7}$$

When the spectral phase error Ep_k is small, i.e. $|\vec{E}_k| \ll |\vec{h}_k|$, $\sin(Ep_k) \cong Ep_k$ and $\vec{h}'_k \cong \vec{h}_k$:

$$Ep_k \cong \frac{|\vec{E}_k|}{|\vec{h}_k|} \tag{10.8}$$

The phase of the error vector is uniformly random over the full circle (as indicated by dashed circle in figure 10.3), but since the error vector phase is statistically independent of the error vector magnitude, the variance of Ep_k remains approximately proportional to the variance of $|\vec{E}_k|$ and inversely proportional to the true harmonic magnitude $|\vec{h}_k|$, for small Ep_k (i.e. high signal-to-noise ratio).

From the foregoing discussion, the following proportionalities exist for the magnitude and phase errors relative to the transform (FFT) length N, the number of quantisation levels L, and the amplitude $|\vec{h}_k|$ of the harmonic with order k.

Dependence on	Magnitude error variance	Magnitude error std dev		Phase error variance	Phase error std dev	
Harmonic order k	independent		(10.9)	independent		(10.10)
Harmonic amplitude ($\|\vec{h}_k\|$) ($SNR \gg 1$)	independent		(10.11)	$\frac{1}{\|\vec{h}_k\|^2}$	$\frac{1}{\|\vec{h}_k\|}$	(10.12)
Number of quantisation levels (L)	$\frac{1}{L^2}$	$\frac{1}{L}$	(10.13)	$\frac{1}{L^2}$	$\frac{1}{L}$	(10.14)
FFT length (N)	$\frac{1}{N}$	$\frac{1}{\sqrt{N}}$	(10.15)	$\frac{1}{N}$	$\frac{1}{\sqrt{N}}$	(10.16)

Table 10.1: Proportionalities for magnitude and phase errors.

10.3 Simulation

10.3.1 Model

A system to generate and analyse harmonics is modelled as shown in figure 10.4. Because the focus for these simulations is on the ADC, analog input circuitry like scalers and low-pass filters

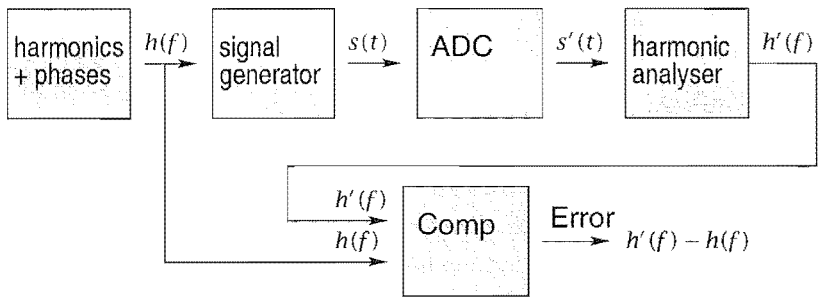


Figure 10.4: Block diagram of the simulation model.

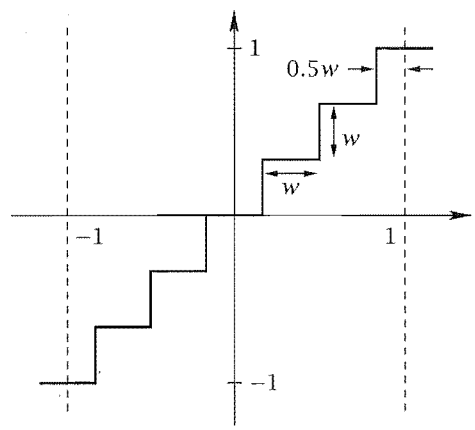


Figure 10.5: The transfer function of the ADC model. For an ADC of B bits, there are $L = 2^B - 1$ steps with a step width and height of $w = \frac{2}{2^B-2}$. Depicted is $B = 3$.

are not represented in the simulation. Their contributions to the frequency and phase errors are constant and can be compensated for in the spectrum.

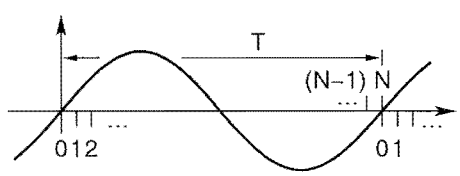
As the first stage in the simulation, a periodic signal $h(f)$ is specified in the frequency domain. It is specified by an amplitude A_0 at a frequency of 0 Hz (DC), and amplitudes A_k and phases γ_k of the fundamental frequency and its harmonic frequencies up to a maximum harmonic H. The time-domain signal $s(t)$ corresponding to $h(f)$ has the formula

$$s(t) = A_0 + \sum_{k=1}^{H} A_k \sin(kt - \gamma_k) \tag{10.17}$$

with H as the highest order harmonic present in the signal. The simulation creates samples of $s(t)$ at regularly spaced times,

$$samples = \sum_{n=0}^{N-1} s(n \cdot \frac{360°}{N}) \tag{10.18}$$

where n is the sample number, and N is the number of points per period T of the fundamental frequency. $\frac{N}{T}$ is the sampling rate. The sample points of $s(t)$ are evenly spaced by $\frac{1}{N}$, with the first point at $t = 0$, and the last point a distance of $\frac{T}{N}$ before $t = T$, or at $t = \frac{N-1}{N}T$. Repeatedly concatenating the sets of sample points results in a continuous signal.



To simulate the effects of quantisation by an analog-to-digital converter, a quantisation function with a characteristic of the form shown in figure 10.5 is used. Intervals in the input are rounded

to the same output value. The number of quantisation steps is determined by the width of the ADC in bits. For a width of B bits, the number of steps L is $L = 2^B - 1$, and the width and height of each step are $w = \frac{2}{2^B - 2}$. To simplify the simulation, it is desirable to keep the range of the input amplitude symmetric around zero. With an even number of steps and one step for the centre, one step remains which cannot be used by a symmetric transfer function. The function as used and as shown in figure 10.5 is odd-valued, and implemented with an input range of ± 1. All amplitudes are discussed with respect to ± 1 (unity).

Within the scope of these simulations, no further digital filtering is performed.

The result of the quantisation is then processed with a harmonic analyser, based on a fast Fourier transform. The harmonic analyser function returns the magnitudes and phases of the discrete Fourier spectrum of the signal being analysed, such that when analysing a signal $s(t)$ with no quantisation noise, it returns the true harmonic magnitudes and phases, $\vec{h}'_k(s(\vec{h}_k)) = \vec{h}_k$ (see figure 10.4). The output of the harmonic analyser with the quantised signal $s'(t)$ is then compared with the true harmonic values which were used to synthesise the input signal.

10.3.2 Simulation Parameters

The simulations are performed with Octave[2], a free open source program similar to Matlab. The model described in the previous section is run with a range of values, in order to examine the magnitude and phase errors in relation to

1. harmonic magnitude and ADC width at a fixed harmonic order,
2. harmonic magnitude and harmonic order at a fixed ADC width, and
3. FFT length and ADC width at a fixed harmonic order.

The effects of ADC width and harmonic order are examined over several bands of harmonic amplitude relative to the amplitude of the fundamental. Each band is given by a lower and an upper amplitude limit. This banding is necessary because the harmonic phase error is inversely proportional to the harmonic amplitude. If amplitudes are randomly chosen between an upper limit and zero, phase errors when the harmonic amplitude happens to be nearly zero dominate the simulation results, concealing the relationship between harmonic amplitude and phase error.

Because the ADC input range is set to ± 1, care has to be taken that the signal does not exceed this limit. The DC value is set to 0. The amplitude of the fundamental is set to 0.5 (i.e. the fundamental is given by $0.5 \cdot \sin(t)$), and giving the harmonics random amplitudes with an upper limit of 0.025 almost always produces signals with peak-to-peak amplitudes $\leq \pm 1$. Therefore, the fundamental is at least 40 times larger than the largest harmonic. The effect of harmonic amplitudes on the resulting error is examined by giving the harmonics random amplitudes within bands. Bands are set in decade intervals starting with 0.025 and are one interval wide, i.e. bands are 0.025 to 0.0025, 0.0025 to 0.00025, and so forth. Each band is examined separately, and for each band each harmonic is independently given an amplitude randomly generated with uniform distribution within the limits of the band.

The sampling rate is specified to the simulating program as the number of samples per period of the fundamental frequency. The absolute value of the sampling rate does not matter because the simulation operates in unit time and frequency. The number of harmonics is set to 128 (including DC), a higher number than is expected to be of interest in real systems. Therefore the number of sample points is set to 256, because there must be at least twice as many samples as the highest harmonic of interest, to satisfy the requirements of the sampling theorem. The Fourier transform length is equal to the number of samples.

Because of the finite numerical precision of the machine's arithmetic when calculating the FFT, the output of the FFT contains arithmetic noise. Harmonics whos magnitude is comparable to or less

[2] `http://www.octave.org/`

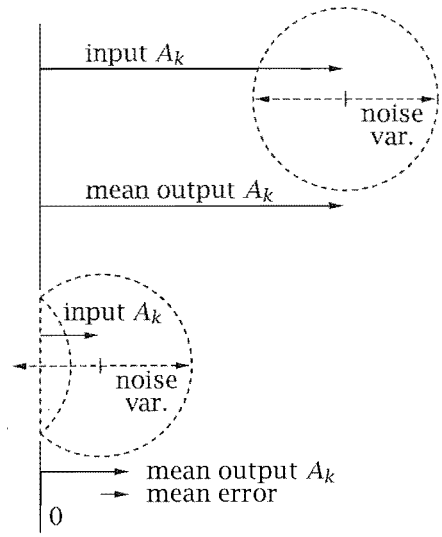


Figure 10.6: Non-zero mean error magnitudes at very low harmonic amplitudes $A_k = |\vec{h}_k|$. If the input magnitude is greater than the noise, the mean output magnitude equals the input; if the input magnitude is less than the noise, the mean output magnitude is greater than the input. Negative values of the noise variance are folded back into the positive side.

than this arithmetic noise level will be swamped by the arithmetic noise. The magnitude values reported for them are close enough to zero within the machine precision and many orders of magnitude smaller than any real harmonic amplitude considered in this simulation, and therefore do not affect results. However, the reported phase values of these noise-harmonics are meaningless and effectively random over the full circle. Because the phases of the harmonic analysis are of particular importance and arbitrary values make a comparison of the phases of input and result difficult or impossible, amplitudes of 0 are never used. Instead of 0, amplitudes of 10^{-9} are used, which is sufficiently small to not affect results. The machine precision is approximately 10^{-16} (64 bit floats).

For each ADC width, harmonic order, or FFT length under consideration, the process of creating a harmonic signal, transforming it, and comparing the result with the initial value is repeated 200 times. For the magnitude and the phase of the error, the mean and standard deviation of these 200 results are calculated and plotted against the variable under consideration.

10.3.3 Results and Discussion

The results of the harmonic magnitude and phase errors as a function of harmonic order at a fixed ADC width are given in figure 10.7 for the magnitudes and in figure 10.8 for the phases. These graphs show that the error is independent of harmonic order, which is consistent with the theory (equations 10.9, 10.10).

Mean and standard deviation of the magnitude error and mean of the phase error are independent of the ratio of the harmonic magnitude to fundamental magnitude (equation 10.11), but only down to a certain limit. The particular limit is related to the quantisation width w of the ADC, or ADC width. When the harmonic magnitude is well above the noise, the mean of the error is essentially zero. For magnitudes close to or below the noise level, the mean increases.

The standard deviation of the phase error decreases linearly with decreasing harmonic amplitude (figure 10.8 bottom), i.e. the standard deviation of the phase error is inversely proportional to the harmonic magnitude. This is in agreement with equation 10.12. Again this is true only up to the limit of noise amplitude. The $2.5 \cdot 10^{-5}$ line is at 60°, not at 100°. These results are understandable because magnitudes can not be negative. Adding random noise values with a mean of 0 to positive values that are smaller or 0 can not result in sums with a mean of 0 (see figure 10.6). This is also reflected in the standard deviation.

Another possible contributing factor to the non-zero mean of small harmonic amplitudes is that when the harmonic amplitudes are less than the quantisation intervals of the ADC, the samples may no longer be uncorrelated with the quantisation levels, in which case the quantisation error is no longer random. Any such correlations can be suppressed by dithering the input signal, as mentioned in section 10.2.

In any case, harmonics with magnitudes this low in comparison to the fundamental are unlikely to be of practical interest in a power distribution system, because their effect on power quality is negligible.

Because figures 10.7 and 10.8 show that the spectral errors are independent of harmonic order, the simulation of the effect of varying ADC width is computed only at a single harmonic order. The 5th harmonic is chosen because it is commonly of interest. All other settings remained unchanged.

The results of the harmonic magnitude and phase errors as a function of ADC width at a single harmonic order and at different harmonic magnitude levels are given in figure 10.9 for the magnitudes and in figure 10.10 for the phases. The mean of the magnitude error decreases with increasing ADC width and is zero for sufficiently large harmonic amplitudes. The standard deviation decreases linearly by a factor of 2 per 1 bit increase in ADC width (doubling of the number of quantisation steps), which is in agreement with the theory (equation 10.13).

The mean of the harmonic phase errors is 0 for sufficiently large harmonic amplitudes. The standard deviation decreases by a factor of 2 per 1 bit increase in ADC width, which again agrees with the theory (equation 10.14).

The bottom graph in figure 10.10 is the most useful one for evaluating the phase recovery performance of a power quality monitor. For example, a system with an 8 bit ADC and 128 point FFT can resolve harmonic phases of unit amplitudes between 0.0025 and 0.025 with a standard deviation of 1.5°, and of amplitudes between 0.00025 and 0.0025 of 40°. A system with a 12 bit ADC has a standard deviation of 0.1° and 2° respectively. (The fundamental has a unit amplitude of 0.5.)

These amplitude values can be related to the total harmonic distortion, THD, which is used as a measure for the overall harmonic content in a signal. It is defined as

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} A_n^2}}{A_1} \tag{10.19}$$

and usually given as a percentage. The THD can be used as a measure for the line losses due to non-linear loads. With an FFT length of 128, up to the 63rd harmonic can be recovered. As an example, the table below shows the THD values that would result from a system with the fundamental amplitude $A_1 = 0.5$ and either 5 or 10 harmonics with an equal amplitude A_n and the remaining harmonics not present.

A_1	0.5	0.5	0.5	0.5	0.5	0.5
A_n each	0.05	0.025	0.01	0.0025	0.001	0.00025
$2 \le n \le 6$, THD	31.6%	15.8%	6.32%	1.58%	0.63%	0.16%
$2 \le n \le 11$, THD	22.4%	11.2%	4.47%	1.12%	0.45%	0.11%

As discussed in section 10.2, the phase noise power is proportional to the inverse of the transform length, therefore the phase noise standard deviation is proportional to $\frac{1}{\sqrt{N}}$. Figure 10.11 shows the phase noise standard deviation as function of ADC width for various FFT lengths between 64 and 1024, and for two different harmonic magnitude bands. The distance between the lines is approximately constant and equal to $\sqrt{2}$, which agrees with equation 10.16.

An increase in FFT length implies a corresponding increase in sample rate. When designing an acquisition system, this can be used to decide on a tradeoff between ADC width, and sample rate and computing power. The computational cost of an FFT is proportional to $N \log N$.

10.4 Discussion and Conclusions

The effects of quantisation noise on the ability to recover phase information have been examined, and simulated with different parameters. The error for recovering phase information is determined by the quantisation noise introduced by the analog-to-digital conversion. The simulations confirm that the phase error is independent of harmonic order; all harmonic orders are equally affected. The magnitude error is also independent of harmonic order, and is independent of the magnitude of that harmonic, for magnitudes significantly larger than the quantisation noise magnitude. The rms phase error is inversely proportional to the harmonic magnitude.

Both magnitude and phase rms error are directly proportional to the size of the quantisation interval, or the inverse of the number of quantisation steps. Both are also proportional to the inverse of the square root of the FFT length. Both factors are intuitively understandable. The rms error in the signal is proportional to the size of the quantisation step, and the total error power is equally divided between each harmonic.

These two relationships are the key factors for assessing the capabilities of a data acquisition system. To increase the precision of the instrument, either or both of the ADC width or the transform length (and therefore the sample rate) can be increased. The graphs in figures 10.10 (bottom) and 10.11 indicate the relationships.

The exact requirements of a data acquisition system depend on the particular application, but a system with 16 bit ADC appears unlikely to be required for power quality monitoring. A 12 bit ADC gives phase errors in the order of a few degrees, and an 8 bit ADC in the order of some tens of degrees, for harmonic magnitudes likely to be of interest for electricity supply applications.

Making use of oversampling is also expected to increase accuracy, but was not investigated within the scope of these simulations. Further simulations could usefully be performed to include oversampling, and the effect of digital filters typically used in data acquisition systems. The use of some type of digital filter is required with oversampling to reduce the data rate by the oversampling factor.

The effects of oversampling on magnitude and phase noise are equivalent to an increase in transform length. Oversampling averages R number of points into each new value, reducing quantisation noise variance by a factor of R, or rms noise by $\sqrt{R}$. Amplifier and ADC non-linearities result in spectral errors whose amplitude is not affected by the oversampling ratio.

The simple quantisation by the simulated ADC produces a flat distribution of noise over the spectral range. Certain types of ADC exist which have a noise spectrum weighted towards higher frequencies, with a 6 dB/octave increase [227]. When used with oversampling, these ADCs push most of the quantisation noise to higher frequencies and out of the frequency range of interest, giving harmonic noise power approximately proportional to $\frac{1}{N^2}$, which is substantially lower than the $\frac{1}{N}$ from simple quantisation.

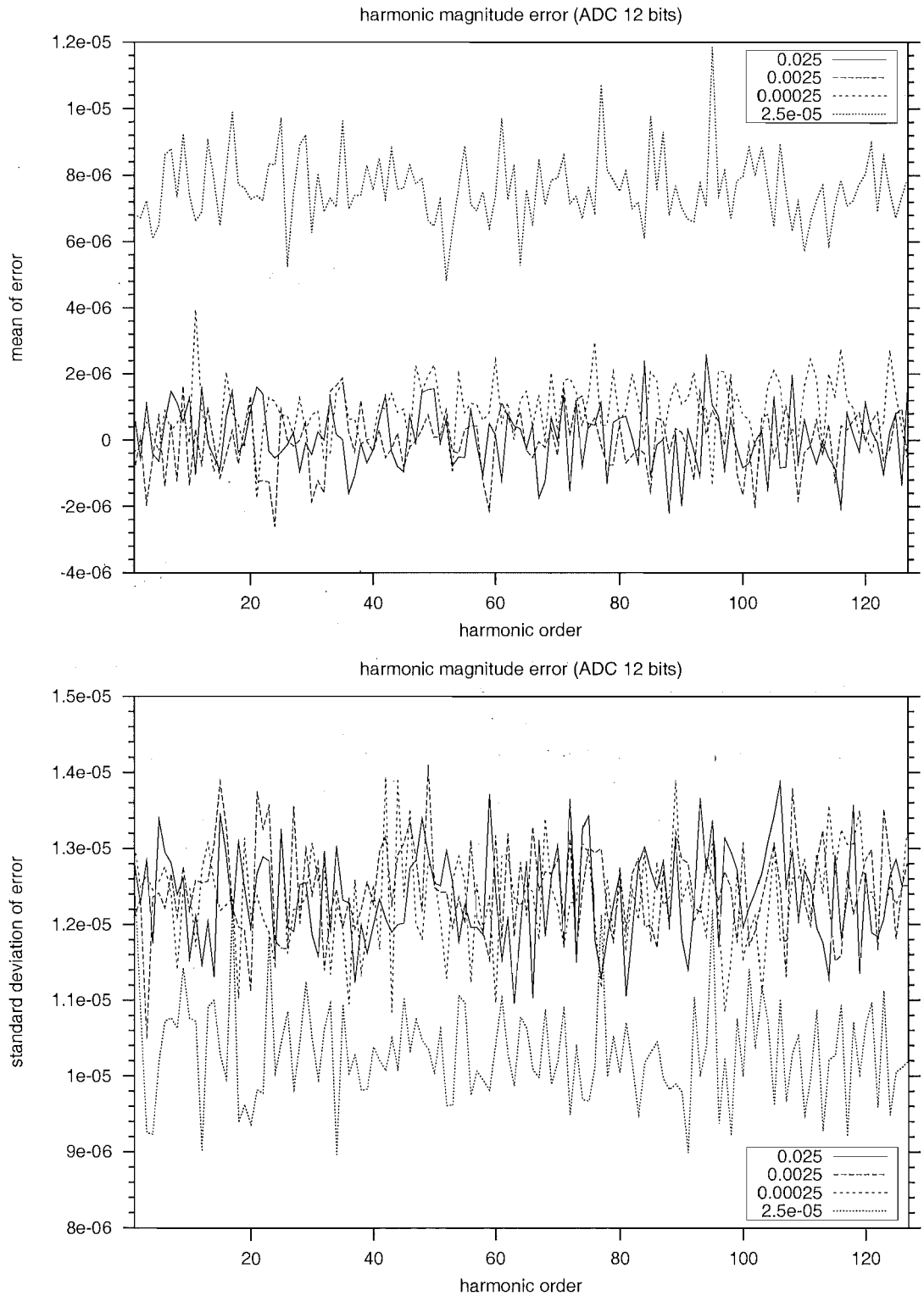


Figure 10.7: Errors in the magnitude of the harmonics as a function of harmonic order, for fixed ADC width and different harmonic magnitude bands. Each line represents a different band of harmonic magnitudes, as shown in the legend in each graph. The top graph shows the mean and the bottom graph shows the standard deviation of the error.

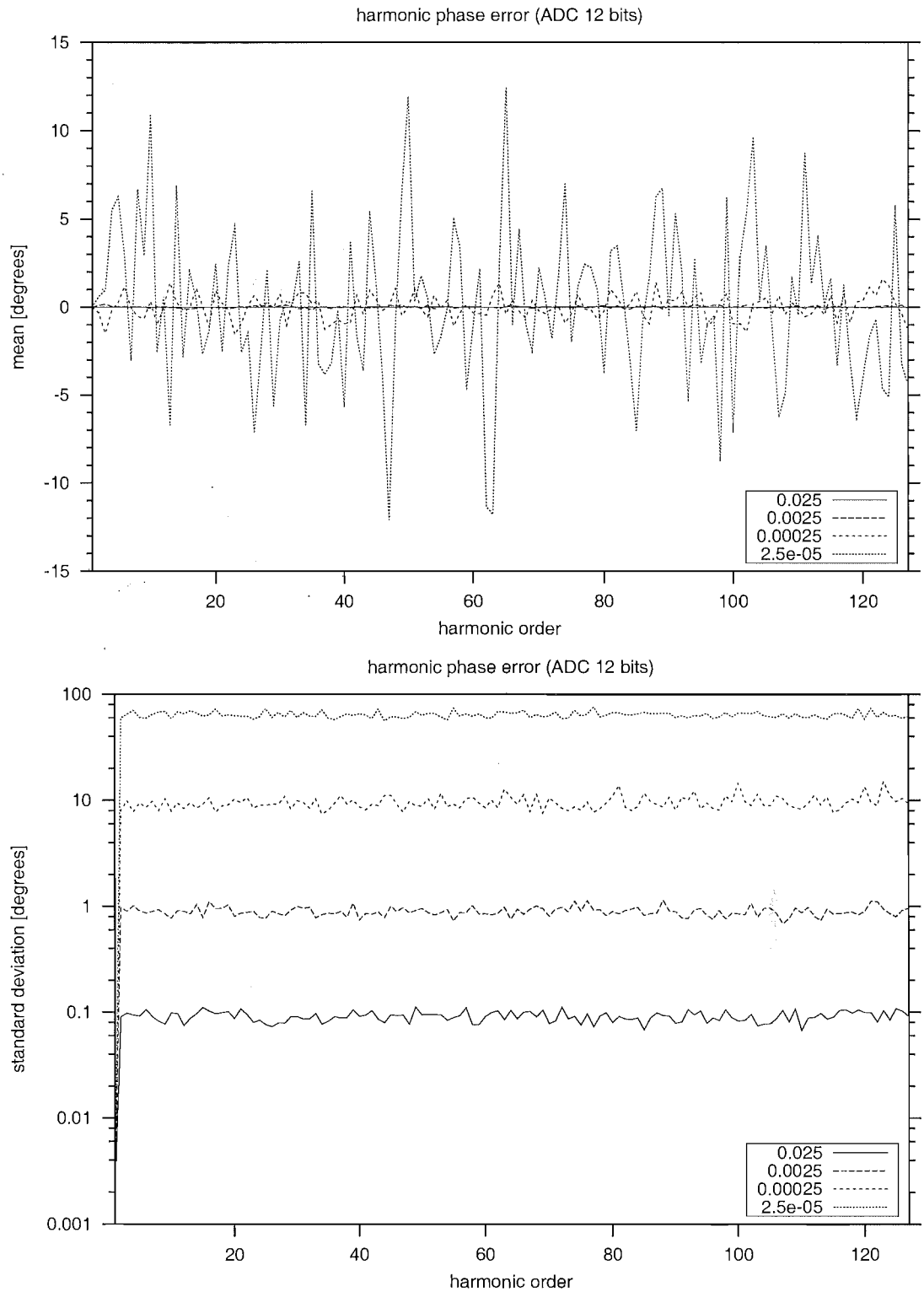


Figure 10.8: Errors in the phase of the harmonics as a function of harmonic order, for fixed ADC width and different harmonic magnitude bands. Each line represents a different band of harmonic magnitudes, as shown in the legend in each graph. The top graph shows the mean and the bottom graph shows the standard deviation of the error.

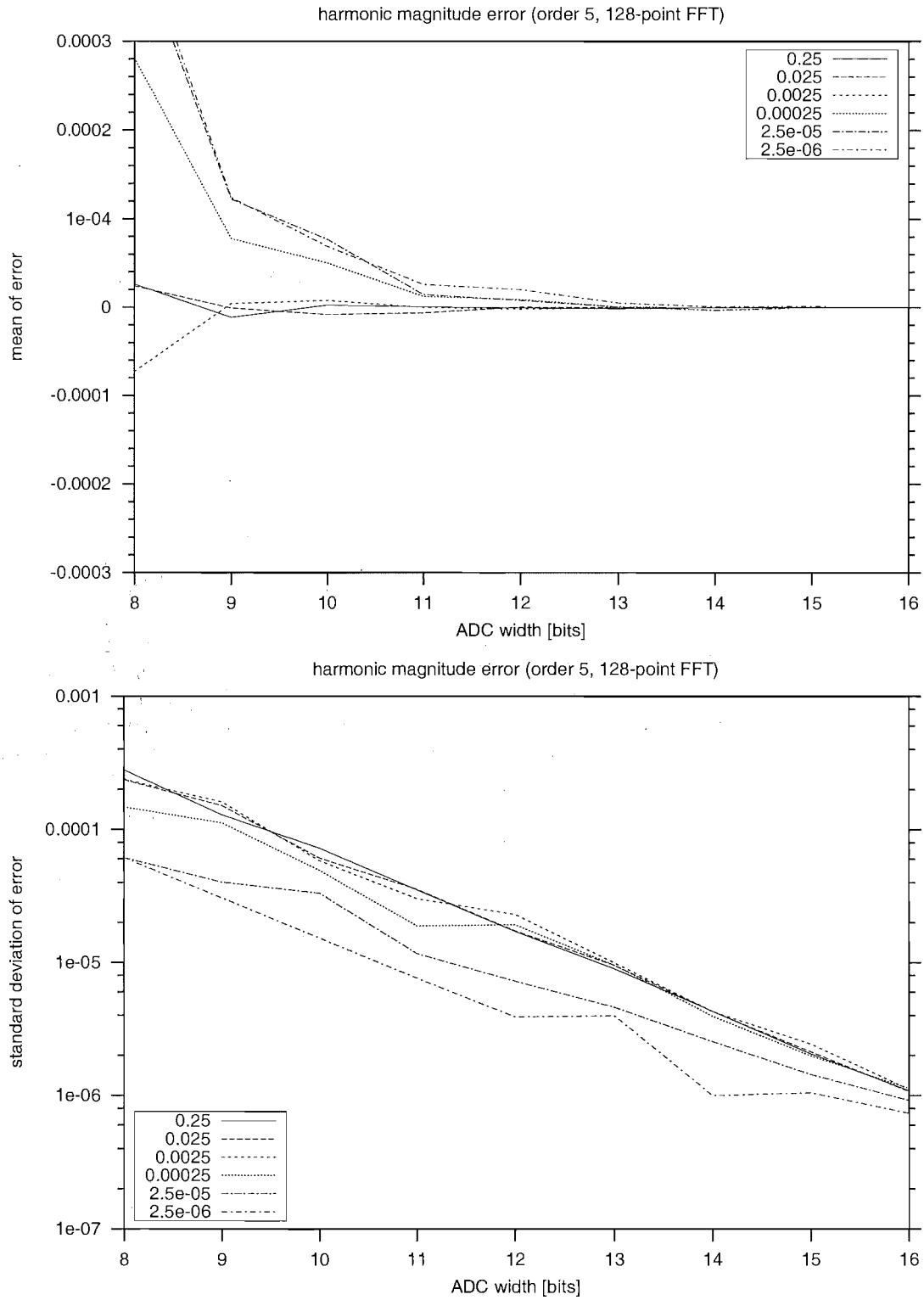


Figure 10.9: Errors in the magnitude of the harmonics as a function of ADC width, for a fixed harmonic order and different harmonic magnitude bands. Each line represents a different band of harmonic magnitudes, as shown in the legend in each graph. The top graph shows the mean and the bottom graph shows the standard deviation of the error.

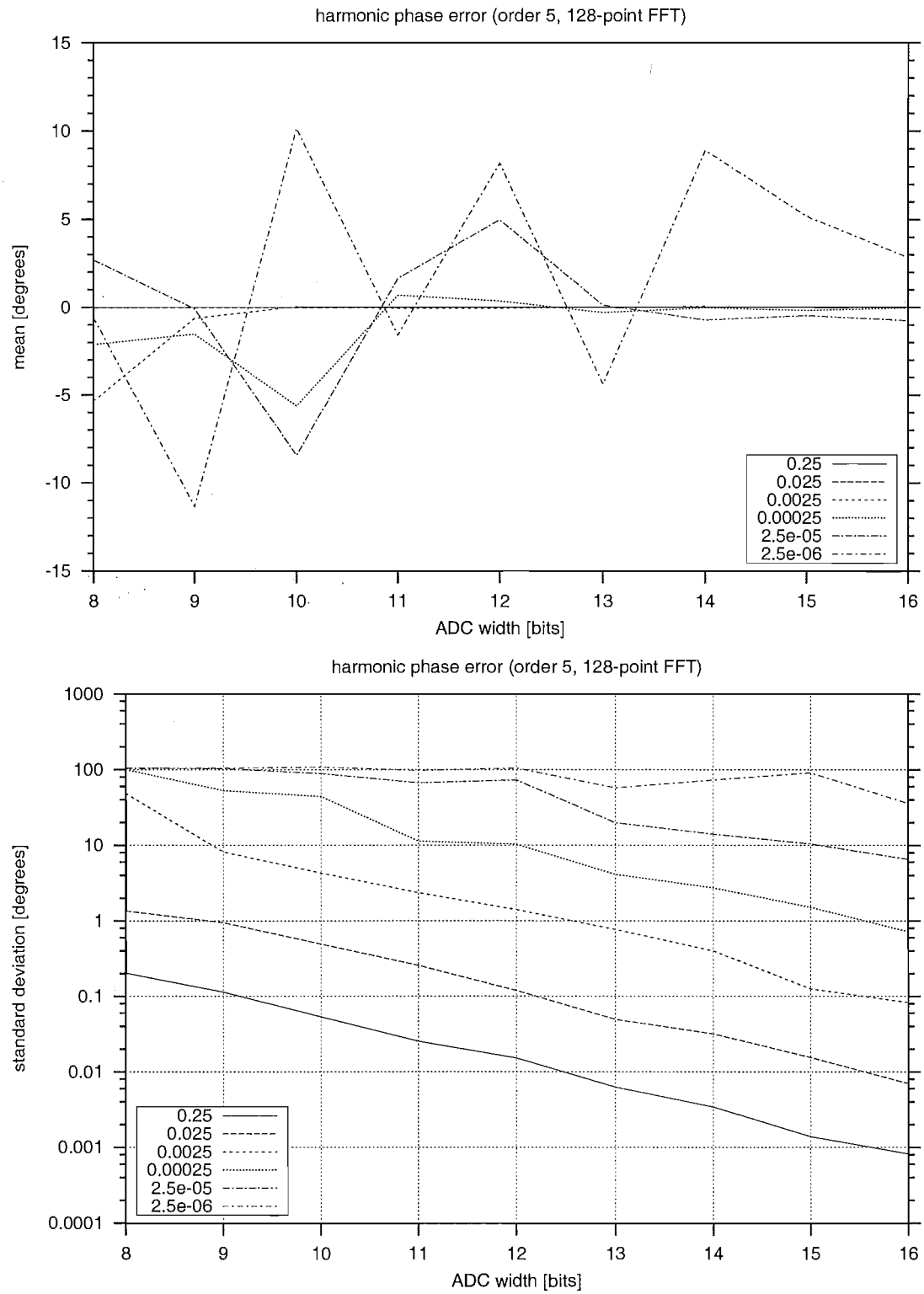


Figure 10.10: Errors in the phase of the harmonics as a function of ADC width, for a fixed harmonic order and different harmonic magnitude bands. Each line represents a different band of harmonic magnitudes, as shown in the legend in each graph. The top graph shows the mean and the bottom graph shows the standard deviation of the error.

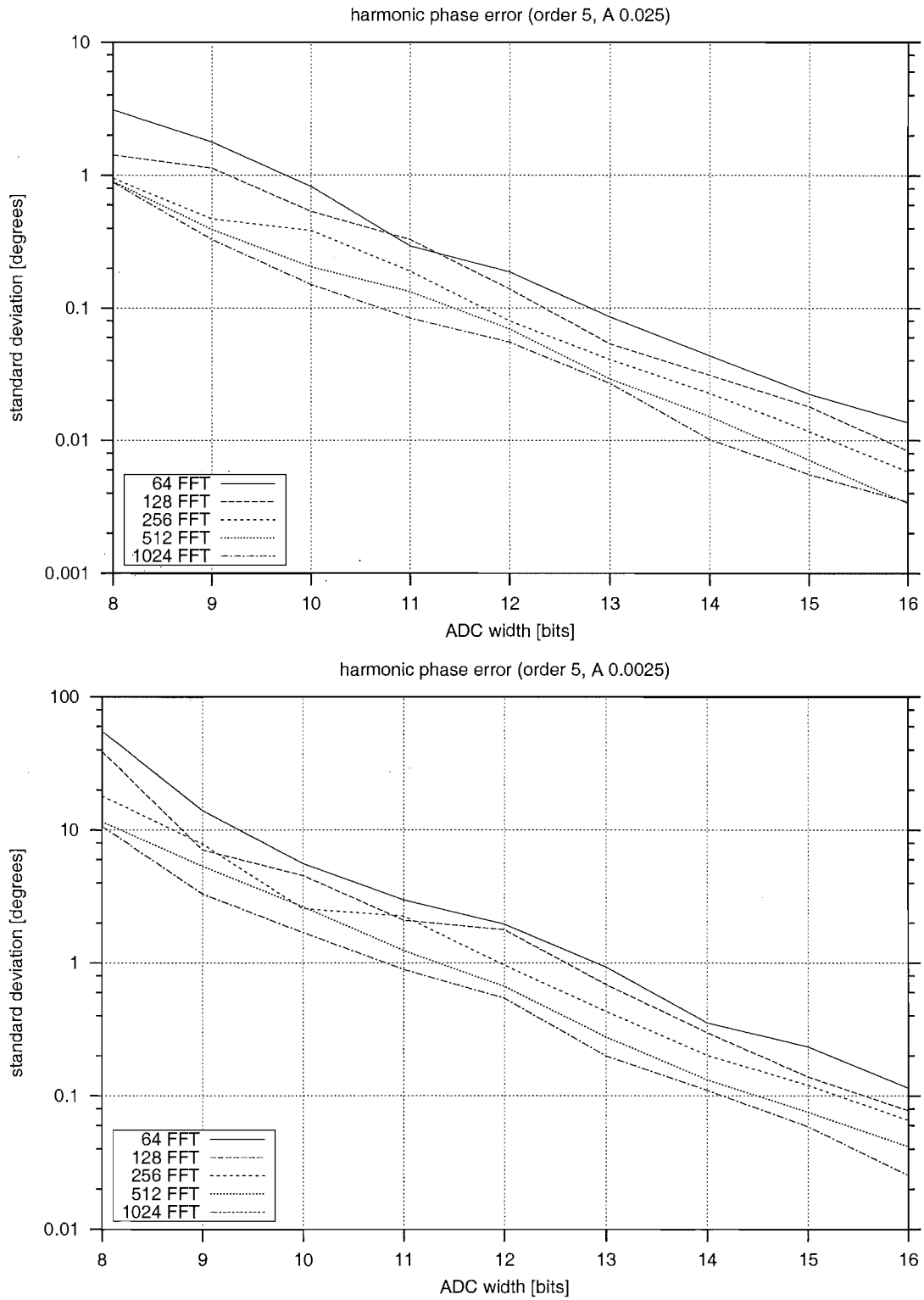


Figure 10.11: Errors in the phase of the harmonics as a function of ADC width, for different FFT lengths and a fixed harmonic order. The top graph is for a harmonic magnitude band of 0.025, the bottom graph for 0.0025.

chapter 11

Advances in Technology

Instruments for power quality monitoring are of interest to both suppliers and consumers. For example, electricity supply networks can be permanently monitored for harmonic power flows, which take up distribution capacity for no real gain. Pollution limits, both legal and agreed, can be observed and policed, and pollution sources can be traced. The instrument's data analysis capabilities should be as versatile as possible, to suit different needs. Ideally, a number of selectable analysis programs which can be commonly expected should be predefined, but there should also be scope for user-defined methods. For fixed installations with predefined scope, instrument versatility can be reduced. A more detailed discussion of applications can be found in chapter 1.

Advances in processor performance bring with them new opportunities for instrumentation systems in terms of size, cost, and bandwidth. Previously, as shown in previous chapters, time-stamping samples with an accuracy of 1 µs required the use of dedicated hardware. This could only be achieved with custom-built hardware, which is time-consuming and expensive.

This chapter examines how technological advances affect instrumentation systems for power quality monitoring in particular. Some advantages are obvious — where one DSP was required per input channel some years ago, a single processor can now handle multiple channels. The biggest savings however can be achieved when it becomes possible to handle the sample clock generation and time stamping by software, possibly with the help of timing registers in the processor or the peripherals.

The main requirements for power quality monitoring are continuous measurement and analysis of voltage and current vectors in power distribution systems. Of particular interest are harmonic components. A sampling rate of 5 kHz is sufficient for the 50th harmonic magnitude (50 Hz system), which does not place a high demand on hardware. The recovery of phase angles for harmonic power flow calculations with an accuracy of 1° – 10° is more demanding, especially for the higher-order harmonics. An even higher demand is placed on equipment when transients are to be examined, but this is not considered in this chapter.

It is essential that the sampling of input values occurs simultaneously on all inputs, so the resulting values, which may for example represent several 3-phase systems in a switch yard or across a power network, can be correlated with each other. The upper limit of allowable time error between inputs is determined by the application, and can be as little as 0.1–1 µs.

Measurements at geographically different locations also require the same accuracy limits, to allow, for example, harmonic power flow calculations. Instruments need to be closely synchronised over large distances. If the instrument is to be used for monitoring, it needs to have sufficient bandwidth in all processing stages to handle a continuous flow of data. In order to optimise data storage and retain relevant data, real-time online analysis is required. Ideally, the ability to preset trigger conditions for storing this data is provided.

Isolation requirements, as needed in a high-voltage switch yard environment, can also put further constraints on instrumentation system design.

A data acquisition system can be broadly described as having the stages

$$sensor \rightarrow ADC \rightarrow processing \rightarrow storage \leftrightarrow analysis$$

plus a suitable control mechanism for the sampling, the complexity of which heavily depends on bandwidth and accuracy requirements. The requirements of the ADC stage with respect to power quality monitoring are discussed in chapter 10. This chapter discusses the key processing issues.

First, the effect of the substantial CPU speed increases in recent years is discussed in section 11.1. This is followed by a description of the key features of PowerPC, Intel Pentium/Xeon and AMD Athlon/Opteron processors and DSPs. section 11.2 then describes a series of code execution time measurements on a number of these processors. section 11.3 discusses factors about time stamping accuracy, sampling rate and synchronisation, that need to be considered for power quality monitoring. Issues for system design, including ADC, front-end and system configuration are discussed in section 11.4. The use of Ethernet to transfer time information is examined in section 11.5. Four different system configurations of varying performance and cost are outlined in section 11.6, and examples are given of commercially-available hardware for building them. The suitability of different operating systems for the systems is discussed in section 11.7, and a general-purpose OS is shown to be an effective option. Finally, the features of some commercial data acquisition systems are outlined in section 11.8 for purposes of comparison and as alternative solutions in some circumstances.

This topic is also examined in two papers, which are reproduced in appendix A. The first one on page 182 looks at the wider issues of system topology, and the second one on page 188 at CPU performance, sample timing and time stamping.

11.1 Processor Architectures

The trend to more complex processors and higher clock speeds is continuing, with both resulting in increased processor performance. Special instructions for signal processing, which were found only in DSPs 10 years ago, are now partially included in desktop processors like Athlon, P4, or PowerPC.

Architectural features like caching, pipelining, and execution concurrency were introduced to improve overall performance. These improvements however had a negative effect on interrupt latency [218]. After continuous speed improvements, latency is in the same order as it was 20 years ago, when compared to improvements in speed over the same period.

A 6502 microprocessor, clocked at 1 MHz in the legendary Apple][computer, in the early eighties (20 years ago) could enter an interrupt service routine (ISR) in 5 cycles: pushing 2 bytes of address and 1 byte of status register onto the stack, and loading 2 bytes of ISR address (program counter) from memory. That is 5 µs. A 80 MHz PowerPC can enter an ISR in 1 µs, and it does require accepting the inconvenience of having to lock the ISR permanently into the CPU-internal cache. Faster PowerPCs are on the market, so the improvement is approximately one order of magnitude. In those 20 years, processor performance has increased by 3 orders of magnitude (a doubling every 2 years [190]). The important detail however is that latency has improved just enough to be able to reach power quality timing requirements.

Improvements to performance have been provided in particular by an increase in memory cache (sizes of 512 kB to 1024 kB are common); pipelining, where the processors can fetch and examine a number of instructions and optimise their order or predict branches; and duplication of internal blocks which can then be used in parallel by different instructions. The complexity has also increased; this often goes with an increase in the instruction set. For example, the vector unit (altivec) in the PowerPC, or the multimedia extension (mmx) or streaming SIMD extensions (sse/sse2) in the Pentium or 3dnow in the Athlon, speed up computations like FFTs considerably.

As well as the poor improvement in latency, a further downside from the point of view of precise timing is the indeterminism of instruction execution times as a side effect of overall performance improvement. The time taken for executing an instruction is to a large extent dependent on whether the instruction is already in the processor's cache. In the case of a cache miss, the instruction will have to be loaded from main memory. The situation is more complex where there is a further cache between the processor's internal cache and the main memory. Processors also have the ability to load a number of instructions and then decide on the optimal sequence of executing them, which also interacts with the compiler being used. With knowledge of the processor's internal architecture and sequencing behaviour, the compiler is able to rearrange the order of instructions while still meeting constraints implied by the program. For the most time-critical tasks the "compiler factor" can be excluded by programming in assembler, e.g. interrupt service routines. The net effect however is that execution times become mostly statistical averages.

A commonly used technique for embedded systems is to lock a code sequence into the internal cache. Some processors, e.g. PowerPC, allow this. Although it reduces the overall available cache size, it guarantees that the code which permanently resides in the cache is executed at maximum speed, and at a known rate.

If the processor is used for time-critical tasks, any non-deterministic execution of CPU instructions becomes irrelevant if the amount of indeterminism is less than the timing accuracy required. Sections 11.3 and 11.4 show that this now appears to be the case with modern processors. There is no longer a stringent requirement for using DSPs for power quality computations.

The remaining part of this section introduces a selection of microprocessors which can be used to build instrumentation systems, and their main characteristics.

11.1.1 Power PC

The PowerPC architecture was designed by IBM and Motorola around 1990. It is a RISC (reduced instruction set computer) design with simple and modern features, and a low power consumption. The RISC design implies that the instruction set is relatively easy to understand and use. The register layout is straightforward: there are 32 integer and 32 floating point registers, all with identical functionality, plus auxiliary registers which every processor seems to have.

In the 1980s, an analysis of the frequency of use of each instruction by a typical program discovered that the complex instructions were almost never used. It was then argued that all complex instructions should be removed, and the execution speed of the remaining simple instructions be improved instead. As a side effect of this cutdown, die size and power consumption were reduced. Over time, more and more instructions seem to be reincorporated into RISC processors. Nevertheless, the general overhaul of the design leaves a programming model which is easy to work with as a permanent feature.

The PowerPC is used by Apple for their range of desktop and laptop computers, and is commonly used for embedded systems in appliances which require reasonably high computing power, such as laser printers. In laser printers, PowerPC replaced the 680x0 family many years ago (probably early–mid 1990s).

The PowerPC provides multiply/add instructions which speed up signal processing computations. Such instructions were previously a prerogative of DSPs. 64 bit versions of the CPU are now available, and multi-processor systems are supported. Also available are versions for embedded systems, which typically have several peripherals integrated on the CPU chip. IBM makes a range of PowerPCs with integrated serial and Ethernet interface, SDRAM controller, and PCI bus controller. Effectively this is a computer-on-a-chip, and is ideal for ground-up designs of flexible and extensible instrumentation equipment.

Motorola also has a range of PowerPC chips with integrated peripherals for embedded systems, centred around a G2 core. The G6 is currently planned so these chips are not the state of the art. They are still fast enough for most power quality applications and do have the advantage of a low power consumption.

The PowerPC is well-supported by development systems and operating systems. Virtually every real-time OS has a version for PowerPC, and the support by Linux is probably almost as good as Linux's support for the x86 family.

11.1.2 Intel Pentium/Xeon and AMD Athlon/Opteron

The design of these processors originated approximately 25 years ago with Intel's 8086, a 16 bit CPU. The first IBM PC, still the dominant low-end computing platform today, used the 8088, a smaller 8 bit version of the 8086. Intel later renamed the platform to Pentium, followed by a number, for trademarking reasons. The equivalent processor from AMD is called Athlon, followed by a designation to indicate speed. Xeon and Opteron are 64 bit versions from Intel and AMD respectively. Multi-processor capable versions of both the 32 bit and the 64 bit models are available, from both manufacturers.

A significant disadvantage of the x86 family is that its basic design dates to the early days of microprocessors, but cannot be updated for reasons of backwards compatibility. For example, the family uses a register model of a single "accumulator" for arithmetic operations, which has not been repeated since the early 1980s. The low number of registers reduces performance and causes difficulties for programmers. Obviously, these processors have still benefited from technological advances, but they still require a relatively large number of transistors to achieve their current level of computing power. Partly because of the large number of transistors, some of these chips have a power consumption of 120 W — PCs are equipped with temperature monitors and automatic shutdown, because if the CPU fan fails, the chip will suffer thermal meltdown within seconds. A key reason for the family's continued development is binary compatibility with legacy software. For any new designs not requiring this legacy compatibility, e.g. laser printers or internet routers, alternative families are usually chosen instead.

However, despite of those disadvantages the x86 family represents excellent value for money, due to their production in large volume. Software support for these processors is also excellent due to their ubiquitous nature. Every company which makes a real-time OS has this OS available for x86. Microsoft's general-purpose OS runs on these processors, and there is a reduced version available for embedded systems. Linux is primarily developed on x86, and there are also real-time versions and versions for embedded systems. A standard PC can be used as development system, providing a cost-effective way for almost all phases of product development.

11.1.3 DSPs

Digital signal processors are primarily intended for certain numerical computations on comparatively high volumes of data. Their instruction sets have always been optimised to allow efficient implementation of numerical algorithms. In the early years, floating point models had instructions to operate on floating point numbers, when general-purpose processors required an additional and optional co-processor to make these instructions available. Later, when the co-processor was integrated into the main chip, DSPs had more complex instructions like multiply-and-accumulate added. Usually, the same operation was faster on a DSP. Floating-point operations did not seem to have received the same level of attention with general-purpose CPUs (e.g. x86, sparc). However, it also has to be said that DSPs used to tend to trade a large mantissa for speed. The C3x/C4x from Texas Instruments for example were strictly 32 bit only.

To keep the device count down, peripherals like timers and ports are integrated to a varying degree. DMA is a standard feature in the higher-end models. The main DSP characteristic is summarised as: *"DSPs are optimised to move data quickly from a peripheral to the DSP core. ... Many DSPs don't provide instruction-cache, but include high-speed on-chip memory that supports efficient program execution."* [119].

In general, it can be said that DSPs are specialised and made for a particular purpose. Outside of embedded systems for which "cheap" and "small" are the key factors (like cellphones or modems),

they are only used as an add-on, a co-processor to a main processor. This is the case with the widely available DSP cards for PCs. There are no computers as such made which are based on a DSP, nor would this be desirable. Hence there are no full operating systems either, only basic limited kernels which serve specific purposes.

The design of general purpose CPUs has been influenced by the requirements of compilers, high-end operating systems, and the C language for creating those operating systems to run on these CPUs. Steve Underwood writes about Motorola's 56k DSP family *"... [doesn't have] an orthogonal general purpose instruction set, because that's not what DSPs are made for. ... the C compiler doesn't understand how to make use of the special DSP qualities of these chips. Since it's those very qualities you are normally using the chip for, using C makes limited sense."* [221]. The implication that C compilers are not useful is overstated. There are C compilers for every DSP, and there obviously is a market for them. It does however highlight that generating good machine code from C is difficult because of limitations inherent in DSPs, and that the resulting code is likely to be relatively slow. This may mean that time-critical parts of the software have to be manually written in assembler.

One possible solution for embedded applications requiring more substantial OS support as well as DSP functionality is to use two processors, and integrate them into the same chip for cost-effectiveness. For example, such a dual-CPU chip, an ARM with integrated DSP, is used by HelloSoft in their VoIP (i.e. internet) telephones [53]. An embedded Linux runs on the ARM, with signal processing functions being offloaded onto the DSP.

In the meanwhile, attempts to port Linux to the Texas Instruments TMS320DM64x and the Analog Devices Inc.'s Blackfin have been successful[1]. In view of the aforementioned discussion about the suitability of DSPs for running general-purpose operating systems, the performance and commercial applicability of doing this will have to be evaluated.

Over time, features were copied between DSPs and general-purpose CPUs, and the distinction has, to some extent, become less pronounced. The Texas Instruments 6000 and Analog Devices blackfin families have more powerful instruction sets geared towards compiler and OS support. While the blackfin uses a RISC approach which should make compiler support straightforward, the 6000 is a VLIW (very long instruction word) design requiring very specialised new compiler technology.

In any case, making use of a DSP in a customary hardware design requires expensive development systems and has drawbacks with respect to software development and run-time efficiency. Development systems for general-purpose CPUs are more frequently used by application software developers than system programmers. As a general rule, DSP software development systems are only used by system programmers and the tools are not as advanced and bug-free.

Making use of another CPU with a DSP as co-processor is too complicated and expensive. As demonstrated in the remainder of this chapter, a single processor provides sufficient processing capabilities for a medium-sized power quality monitoring system. The DSP simply becomes superfluous. Development systems and compilers are more advanced for general CPUs. The compiler offered by TI for the C3x/C4x DSPs for example was lacking in key aspects, as were competitive products. There are first-class compilers available for e.g. the x86 and PowerPC families (and one of the best is even free of charge). Choosing a general-purpose CPU is more flexible and offers a much wider range of available off-the-shelf hardware and software.

11.2 Benchmarks

One of the major computational tasks for a power quality monitoring system is to perform FFTs. A benchmark has been performed for a number of common processors. The fftw library from `http://www.fftw.org/` has been used [43–45]. It is available for several architectures and contains a benchmarking program. A special characteristic of fftw is that it performs a number

[1] `http://deviceforge.com/news/NS9674515032.html`

of tests at runtime on the target system to construct code which runs fastest under actual conditions. This is a clever way to deal with the unpredictability of modern processors with respect to execution speed. Before any FFT can be computed, the code for it needs to be generated once, which takes for example the number of points as argument. For power quality instrumentation this does not create a problem because the type of FFT for any given task is constant once the type of measurement and evaluation have been decided upon.

All tests were performed with 128 data points and with complex numbers, for both single and double sized floats. 128 points were chosen because it is the number as used for CHART. Processor optimisations were enabled where available. Some results are included with these processor features unused, for comparison. The Apple powerbook was operated under its native Mac OS 10 (Darwin), which uses a Mach 3.0 kernel with operating system services based on 4.4BSD. All other machines were running Linux with a kernel around 2.4.20 and with the C compiler gcc 3.2 or 3.3. Threading was disabled on those computers with two CPUs, unless otherwise noted, to get results which are valid for a single CPU. A "single" float is 4 bytes long (32 bits), and a "double" float 8 bytes (64 bits).

The hardware which was used for the FFT benchmarks is briefly described below.

G4	Apple powerbook G4, 1.25 GHz, L2 cache 512 kB, bus speed 167 MHz, Mac OS 10.3.2, model M8981LL/A.
P3	No-name desktop PC with P-III (Katmai) CPU, 450 MHz, cache 512 kB.
P4 mobile	Compaq laptop with P4 mobile CPU, 1.8 GHz, cache 512 kB.
P4	No-name desktop PC with P4 CPU, 2.4 GHz, cache 512 kB. This is not a hyper-threading model.
Athlon 1800+	No-name desktop PC with dual AMD Athlon XP 1800+ CPUs, 1.53 GHz, cache 256 kB.
Opteron	No-name desktop PC with dual AMD Opteron CPUs, 1.8 GHz, cache 1024 kB. During the FFT tests, 4 other CPU-intensive processes were running, creating a CPU load of 200%. When the benchmark was repeated on an unloaded machine, the results were the same.
Xeon	Dell PowerEdge 2650 Server with dual Xeon CPUs, 2.4 GHz, cache 512 kB.
Ultra SPARC	Sun Enterprise 450 (4 X UltraSPARC-II 296MHz). During the FFT tests, 2 of the 4 CPUs were otherwise occupied.

The results are shown in table 11.1. The benchmark program was run several hundred times for each combination of CPU and options, and the results combined into a single figure. Sometimes the time reported by the benchmark was significantly above average, those values were discarded. As with any benchmark, results depend on many parameters and should always be treated with due caution. Although these benchmarks were performed while the processors were otherwise unloaded (except where noted), repeating them while other users were running their programs at the same time did not produce different results.

A G4 processor is able to perform this FFT in $\approx 4.5\,\mu s$, and there is no speed difference between single and double floats. Making use of the processor's vector unit makes it more than twice as fast, but the unit is only available for single floats.

A Pentium 3 at 450 MHz needs $\approx 14\,\mu s$ when using the processor's optimised instructions, there is also no difference between single and double floats. This PC is current as of 1999, similar hardware is now cheaply available for embedded systems. It is interesting to note that enabling the use of additional instructions (sse, sse2) does not seem to have any effect. Perhaps they are used anyway when found.

A Pentium 4 (mobile version) at 1.8 GHz needs a little under $2\,\mu s$ for single floats, and a little under $3\,\mu s$ for double floats. There are no more instructions in the P4 than there are in the P3 (if

CPU	single	double	optimisations	comment
G4 1.25 GHz	4.4 µs	4.4 µs	fma	
	2.0 µs	n/a	fma, altivec	
P3 450 MHz	14.0 µs	14.4 µs	—	
	13.9 µs	14.1 µs	sse	
	13.9 µs	14.2 µs	sse2	doesn't have sse2
P4 mobile 1.8 GHz	1.93 µs	n/a	sse	
	n/a	2.89 µs	sse2	
P4 2.4 GHz		3.32 µs	—	
	2.83 µs	3.30 µs		
	1.52 µs	n/a	sse	
	n/a	2.04 µs	sse2	
	1.51 µs	n/a	sse, threads	not hyper-threading
	n/a	2.22 µs	sse2, threads	not hyper-threading
Athlon 1800+ MP (1.53 GHz)		3.30 µs	—	
	2.29 µs	3.30 µs		
	2.34 µs	n/a	sse	
	n/a	3.33 µs	sse2	doesn't have sse2
	1.27 µs	3.30 µs	k7 (3dnow)	
	1.27 µs	3.36 µs	k7 (3dnow), threads	dual-processor
Opteron 244 MP (1.8 GHz)		2.75 µs	—	
	1.83 µs	n/a	sse	
	n/a	2.75 µs	sse2	
	1.07 µs	2.76 µs	k7 (3dnow)	
	1.07 µs	n/a	k7 (3dnow), sse	
	n/a	2.75 µs	k7 (3dnow), sse2	
	1.07 µs	2.76 µs	k7 (3dnow), threads	dual-processor
Athlon64 3200+ (2 GHz)	2.41 µs	2.20 µs	—	
	1.19 µs	n/a	sse	fftw 3.1
	n/a	1.95	sse2	fftw 3.1
	0.90 µs	n/a	k7 (3dnow), sse	fftw 3.1, 32bit code
	n/a	2.43 µs	k7 (3dnow), sse2	fftw 3.1, 32bit code
Xeon 2.4 GHz MP		3.33 µs	—	
	2.94 µs	3.35 µs		
	1.40 µs	n/a	sse	
	n/a	2.17 µs	sse2	
	1.46 µs	n/a	sse, threads	dual-processor
	n/a	2.15 µs	sse2, threads	dual-processor
ultrasparc 296 MHz	14.06 µs	13.86 µs		

Table 11.1: Execution times of a 128 point complex FFT on various processors, using the fftw library and its bench program. On dual-processor machines (MP), threading was turned off unless noted.

CPU	points	single	double	optimisations	comment
P3:	128	8.02 µs	14.3 µs	sse / —	
	256	16.6 µs	32.1 µs	sse / —	
	512	34.8 µs	74.6 µs	sse / —	
	1024	92.7 µs	205 µs	sse / —	
Athlon:	128	1.27 µs	3.34 µs	k7 / sse2	
	256	3.84 µs	7.15 µs	k7 / sse2	
	512	8.04 µs	16.25 µs	k7 / sse2	
	1024	17.3 µs	35.3 µs	k7 / sse2	
Opteron:	128	1.07 µs	2.75 µs	k7 / sse2	
	256	3.16 µs	5.86 µs	k7 / sse2	
	512	6.65 µs	13.5 µs	k7 / sse2	
	1024	14.4 µs	29.0 µs	k7 / sse2	

Table 11.2: Execution times of complex FFTs with varying number of points, using fftw. The hardware is as for table 11.1.

anything, there are fewer), and the increase in speed fairly matches the increase in clock speed. P4-based boards for embedded systems can now be obtained easily.

The speed of an Athlon XP is comparable to a P4. The P4 is a little faster for double, and the Athlon is about a third faster for single. The 3dnow processor instructions seem to have no effect for double.

An Opteron 244 at 1.8 GHz is only fractionally faster than a P4 at the same clock frequency, despite its 64 bit architecture. The speed difference could possibly be explained by the 4 times larger cache size of the Opteron. However, Athlon XPs and Opterons have additional instructions over the P3/P4, which give a significant advantage of an additional 2/3 speed improvement. Unfortunately, this is only effective for single floats. Considering its 64 bit architecture, the Opteron's performance for double floats is disappointing. However, if only single floats are required, which is generally the case for power quality applications, it is the fastest processor available by some distance.

A Xeon at 2.4 GHz is comparable with an Opteron at 1.8 GHz. It is slower for singles, and faster for doubles. The Opteron is much more cost effective.

Threading is a technique to utilise more than one CPU, or duplicated areas inside the same CPU, for the same program. Different parts (threads) of the program are executed on different hardware in parallel. Interestingly, threading does not seem to provide an advantage in this case. In some situations the use of threading even seems to slow the program down a little. Computation of FFTs does lend itself to parallelisation, but perhaps a 128 point FFT doesn't sufficiently benefit to result in an overall speed gain.

Figures for an ageing Ultra SPARC are included mainly because the hardware happened to be available. Its performance is comparable to a P3-450. The Ultra SPARC and P3 are the only processors with identical performance for single and double floats, at least when computing a 128 point FFT.

When applied to a power supply system, a new FFT has to be calculated every 20 ms (50 Hz system). Modern processors are more than three magnitudes faster than that. All contemporary processors complete the task in under 5 µs. An outdated P3-450 is still under 15 µs. These execution times need to be multiplied by the number of channels the instrumentation system is able to handle. Even if the number of points is increased, computation time remains within acceptable limits. As table 11.2 shows, a 1024 point complex FFT on a P3-450 takes about 200 µs at double precision. Neither double precision nor a 1024 point transform length are needed for power quality applications.

Therefore, the question of which processor is best for use in a power quality monitoring system should not be answered by processor speed at all, because they all provide adequate performance.

Questions of cost and the availability of development systems should be the most important factor, although the cost of the development system proportionally reduces with larger production volumes. The P3 should not be ruled out, because it also provides adequate performance, and might be available on boards for embedded systems with a low-power version. For a desktop system, however, the speed differences are significant.

There should be ample scope for handling 48 channels, a modern operating system with data I/O functions for storage and network communications, and the generation of sampling clocks and time stamps on just one processor. A 64 bit CPU is unnecessary, and a 32 bit Athlon, P4, or PowerPC is sufficient. The best cost/performance ratio is offered by the Athlon, but ultimately other factors like power consumption or availability of off-the-shelf processor cards also need to be considered when designing instrumentation systems.

11.3 Sample Timing Issues

The first step for any data acquisition system is to convert an input value to a digital representation. This sampling process is initiated by a signal to the ADC (analog-to-digital converter) stage. Depending on the application, a data acquisition system may have many input channels, each of which has an ADC. Synchronous measurements require that the sampling on each input occurs at precisely controlled instants. The maximum allowable uncertainty for this depends on the particular measurement being undertaken, and is often much shorter than the time between samples.

For distributed power quality monitoring or harmonic state estimation, the time stamping accuracy of the sampling typically needs to be in the order of 0.1–1 µs. Harmonic power flow calculations compare phase angles at different nodes of the distribution network. 1° at the 50th harmonic is approximately equivalent to 1 µs. The samples need to be time-stamped with an accuracy better than 1 µs.

For a 50 Hz system, a sampling rate of 5 kHz would be sufficient to capture the 50th harmonic, but it is advantageous to use a power of two of the fundamental, e.g. 6.4 kHz (128 points at 50 Hz, no oversampling).

Synchronisation of sampling, or channel coherency, is achieved by clocking all ADCs with the same signal. The ADCs must have a known and constant time after which the result appears at the outputs. The following tasks need to be performed:

- sample clock generation
- sampling at a multiple of the fundamental frequency
- time stamping

Sampling at a multiple of the fundamental frequency is not strictly necessary, but does significantly reduce filter and FFT computational requirements by eliminating windowing issues.

It should be possible to perform all these 3 tasks by utilising a 32 bit counter register provided by the CPU. The counter is free-running with a suitably high frequency, about 50–100 MHz (or the clock frequency of the CPU), and pre-loaded with a value which, when decremented to zero, denotes an interval between samples. When the counter reaches zero, a sampling signal for the ADCs is generated, the counter is reloaded, and the ADCs are read.

The sampling frequency is adjusted by changing the initial counter value. The actual mains frequency can be measured by feeding a mains-derived and low-pass filtered zero-crossing signal into an interrupt input of the CPU. The mains frequency can be derived from the initial counter values and the number of times the counter has reached zero. Calculating the mains frequency is not time-critical and should be performed outside the interrupt service routine. Using hardware counters for generating a mains-synchronised sampling clock is described in [101]. Obviously, any system should also alternatively allow for sampling at a fixed rate.

Time-stamping is performed by generating another interrupt by the one-second output of the time receiver. In the interrupt service routine, the above counter needs to be read and the value stored. This allows the determination of the number of counter decrements per second, which can be used when reading the ADCs to work out the number of counts which have elapsed since the beginning of the second.

Of critical concern is the timing and latency of the interrupts, but a number of factors can be used to advantage. All counter calculations only involve simple integer calculations and need not be performed inside the interrupt service routines. The order of operations inside the interrupt service routines should be considered carefully.

The interval of the 1 PPS signal can be considered constant. The counter frequency will drift, but not change abruptly. This interrupt should be assigned highest priority. If the counter frequency (~ CPU clock frequency) is calculated to have jumped by an unreasonable amount, the result should be discarded for this second.

Likewise, the zero-crossing interrupt may be masked by a higher-priority interrupt for a short time, < 20 µs. The software can cope with this, combined with the assumption that mains periods do not change significantly for single cycles. A simple suitable low-pass or averaging filter should be used as well, which also eliminates jitter generated by noise.

The sample clock interrupt is non-critical. When reaching zero, the counter generates an external clock signal and reloads, without software. The ADCs can be read any time before the next sample clock pulse.

An 80 MHz Power PC CPU has an interrupt latency in the order of 1–2 µs. 400 MHz versions are available, with correspondingly lower latencies. A large part of the latency is deterministic and can therefore be compensated for. An accuracy of < 1 µs can be achieved.

It is therefore possible to perform sample clock generation and time stamping in software with the help of a suitable CPU register, while only placing a small to moderate load on the CPU, leaving capacity for processing of the data. A 1 µs accuracy (class T5) can be achieved with a tight integration of hardware and time stamping software. A 25 µs accuracy (class T3) should be easily achievable with a modern operating system and current hardware, using a high-priority interrupt.

11.4 System Architecture

Power quality monitoring places a number of additional constraints on conventional instrumentation systems. Switch yards, for example, are an electrically noisy environment and can produce potentially large voltage differentials in instrumentation systems. Signal sources are likely to be located many metres from a control room. This makes it impractical to carry signals to ADCs located in control/substation enclosures because of the induced noise. Instead, ADCs should be located close to the signal sources. If signal data is carried digitally to the control room, it is relatively immune to noise and further signal degradation is prevented. The best noise immunity on switch yards is obtained with optical communications, which are available at low cost in the form of 100M Ethernet.

The requirements that have by far the greatest effect on the design of the overall system are the need to place ADCs close to the signal sources, and time stamping accuracy [108]. Previously, the cost of the time stamping hardware (GPS receiver, sample clock generation) and of the signal processing hardware caused the concentration of these functions in one base unit. This necessitated the transmission of the sample clock to the separate ADC units and the return of the data digitally through a noisy environment, as shown again for comparison in the simplified figure 11.1. The remaining part of this section introduces a new design which makes use of technologies which have become available since the design of CHART III. A block diagram of this new design is shown in figure 11.2. Outlines of further designs emphasising different design goals are discussed as well.

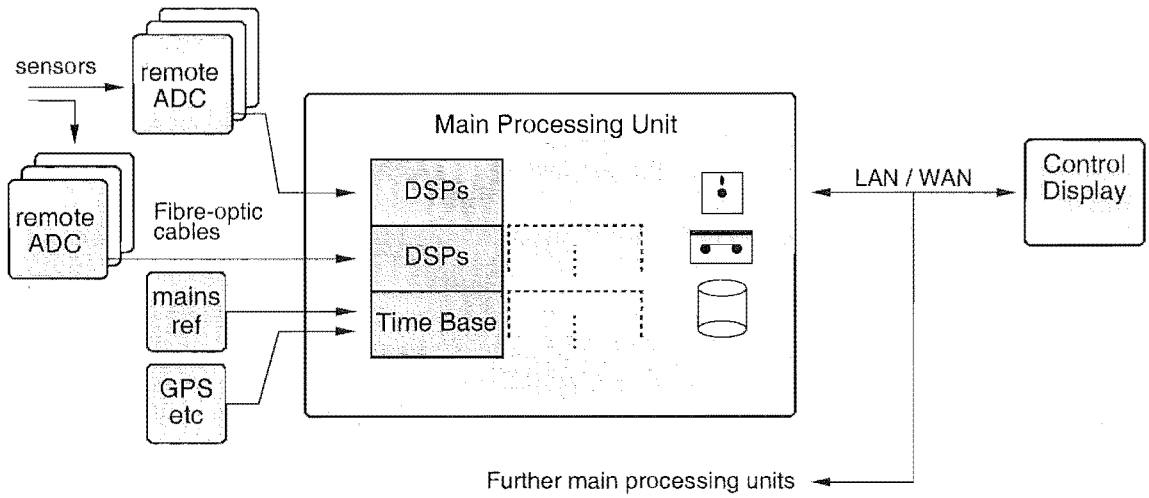


Figure 11.1: A typical configuration of the previously developed CHART instrumentation system, which has a hardware time base.

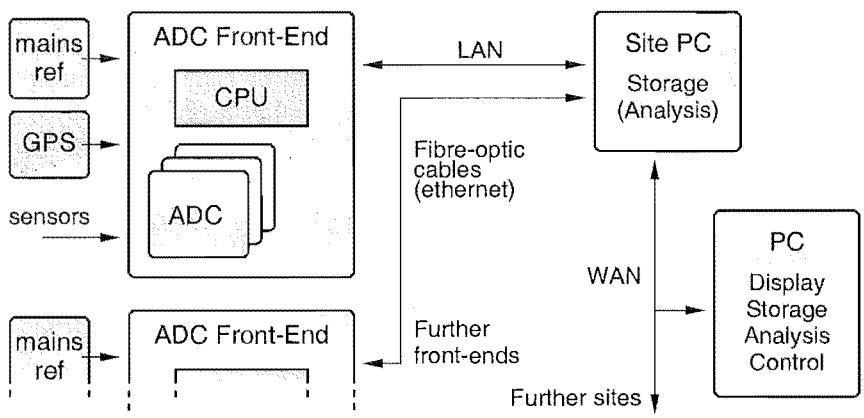


Figure 11.2: A typical configuration of the proposed sampling hardware. The mains-ref unit may alternatively be connected to the site-PC.

For measuring voltages and currents of a 3-phase power distribution system, 6–8 ADC channels are needed. Therefore, to be usable practically, the minimum number of channels in a system is 6. At a sample rate of ≈50 kHz (8× oversampling) and an ADC width of 12 to 16 bit, the resulting 600–800 kbyte/s data rate is well within the capabilities of current CPUs, with computing power to spare.

Oversampling, as in this data rate estimate, increases the bandwidth requirements. More bandwidth is also needed for an anti-alias filter. The advantages of oversampling and filtering are a reduction of computational requirements further on in the processing chain, and a reduction in analog front-end filter component size.

A standard PCI (Peripheral Component Interconnect) bus is clocked with 33 MHz. A 32 bit word can be transferred in one bus cycle in burst mode, which carries an overhead of a few cycles. A single bus access takes approximately 4 bus cycles. This throughput is sufficient for many more than 6 ADC channels. Another research group in the Electrical Engineering Department[2] manages to transfer in excess of 10 Mbyte/s from custom-made ADC cards with DMA to a hard disk with FFT computations on a P3-800 CPU in a CompactPCI system [52].

Versions of the PCI bus with increased clock speed and 64 bit width have been specified, but have not experienced widespread use in desktop PCs. A new combined serial/parallel bus with 2.5 GHz

[2]Senior Lecturer Dr. Michael P. Hayes / Philip Barclay / et al., personal communication

bus clock, PCI-Express, has a significantly improved bandwidth, and is set to eventually replace PCI. This amount of bandwidth is not likely to be needed by power quality instrumentation. PC bus systems preceeding PCI are no longer in common use.

The more processing that can be performed in the front-end, the less bandwidth and computing is needed further on at the central processing and analysis stage. However the particular application will set a limit on how much can be left to the front-end. The tasks of generating the sampling clock and time stamping the samples, as described in the previous section, will need to be performed in any case.

Designing inputs with auto-range hardware has the advantage of effectively increasing the ADC width and allowing a direct connection of a larger range of sensors or other hardware like current or voltage transformers. The disadvantage is that the cost of providing the hardware on the ADC card is significant. The front-end has to provide computing resources for implementing a peak detection of all samples. The input signal is disrupted during a change of range. Whether or not this is an issue depends on the particular application. Programming the system so that the range-selection is fixed is not very different to using a separate input amplifier. The main disadvantage of requiring a programmable input range is that it may effectively preclude using off-the-shelf ADC cards.

If a commercially available ADC card fulfils every other requirement, the need for auto-ranging, and the consequent cost impact, should be re-evaluated. In some circumstances, auto-ranging is not effective, for example where sudden large peaks are to be captured accurately and without being clipped. Auto-ranging is not needed for measuring voltages, or when the approximate current level is known in advance. However, for continuous monitoring of varying current levels, auto-ranging significantly extends the effective ADC resolution.

External pre-scalers may be appropriate for the application. These can be put into small cases and be battery-powered, with a mechanical switch operating on three to four signal lines at a time.

Pre-processing the digitised data before transmission to a centralised analysis unit has several advantages. Depending on circumstances, a suitable noise-reduction or low-pass filter can be implemented. Compensation for certain non-linearities of the sensor will make post-processing of data easier. This can be sped up by using simple pre-computed look-up tables. A 16 bit ADC and 32 bit floating point numbers translate into a 256 kbyte table size, which is small by today's standards. While it is not essential to compensate at the ADC front-end, doing so presents filtered and corrected data as output of the ADC front-end, which reduces complexity and processing at later stages.

At the minimum, it is desirable that the ADC front-end provides at least slightly noise-filtered data which is time-stamped with sufficient accuracy and corrected for sensor non-linearities.

11.4.1 Front-End Design

It would certainly be possible to custom-design all the necessary hardware in a way similar to CHART III, but it would likely be uneconomic.

Instead of making custom hardware which would likely be uneconomic, front-end CPU systems can be based on one of the commercially available single-board computers. Besides a CPU and memory, these boards often provide at least one PCI slot and various peripheral interfaces, including RS-232C, Ethernet, or IDE hard disk. Intended for use in embedded systems, the form factor is comparatively small, and a low-power version of the CPU is used. CPU throughput is therefore likely to be a little lower than in top-end desktop PCs. In other words, "a laptop with PCI slots and no case" is a reasonable description of the desired features. Some of the commercially available hardware suitable for use in a monitoring system is discussed in section 11.6.

Using, in essence, a PC for this ADC front-end provides further benefits. Single-board computers are commercially available at comparatively low cost. Most of the software development can be done on a PC. Software development systems are much more advanced for PCs and standard tools

can be utilised. Development for embedded systems is intrinsically more difficult and requires expensive specialised tools.

Some custom hardware will need to be interfaced to this CPU, which is easiest when implemented as a PCI bus card. The custom hardware comprises the analog input stages, scale adjustment for auto-ranging, and ADCs, once per input channel, as well as those parts of sample clock generation and time stamping circuitry which can not yet be implemented in software. It is desirable that all required custom hardware is integrated on one or more PCI cards. If more than one card is required, all cards should be identical in design. Depending on the particular components used, it is possible to fit everything into a standard small form factor PC enclosure, or whatever enclosure is typically used by the bus system chosen for the front-end.

For measurements involving more than one geographical location, accepted practise is to use a commercially available satellite time signal receiver as time base, e.g. GPS, GLONASS, or the planned Galileo. Some differences between these systems exist, for example GPS does not work indoors, whereas indoor operation is designed into Galileo. In the future receivers may be available which can utilise more than one satellite system for increased reliability, or better political independence.

These time signal receivers supply coarse timing information (1 s and up) via a serial RS-232C interface, and a digital one-pulse-per-second (1 PPS) output accurately ($< 1\,\mu s$) designating the start of a second, which must be interfaced to the time-stamping circuitry. If the time-stamping is implemented in hardware, the 1 PPS signal will have to be supplied to all PCI cards via interconnecting cables. The propagation delay over these cables must either be identical at all locations or small enough to be insignificant.

Flash memory is now available in capacities which are high enough to store the operating system for the front-end CPU, all the application software, and correction tables for a large number of sensors. The system needs to boot from this flash memory. Some single-board systems provide an IDE hard disk interface and/or a compact flash card interface, which any operating system should be able to access easily. The IBM/Hitachi Microdrive is a tiny harddrive which is physically and functionally identical to a compact flash card and can be used as a larger-capacity replacement.

Power supply issues need to be resolved. Batteries can be used for continuous operation up to a certain time limit. For permanent operation, a suitable isolated power supply must be made available.

Environmental issues in terms of temperature, moisture, and electrical noise have to be considered. Unless the equipment is only expected to be used strictly indoors, it needs suitably rugged and moisture proof enclosures. Sealed rugged plastic cases are easily available. Shielding must be sufficient for the electrical noise present in switch yards or substations for reliable operations of digital systems. Appropriate attention needs to be paid to the shielding and noise-immunity of 16 bit ADC circuits or the effective resolution will be reduced. Although an off-the-shelf x86-based computer is by far the most cost-effective computer system available, whether it performs satisfactorily under these circumstances needs to be tested.

The two most viable candidates for a bus system are CompactPCI and PC/104. PCI[3] is ubiquitous in desktop computers. The compact version is electrically identical, but uses a smaller connector and an approximately Eurocard form factor ($100 \times 160\,mm^2$). The base version operates at 32 bits and 33 MHz. 64 bit versions are common on high end boards, which may also operate at 66 MHz. PC/104 uses a $90 \times 90\,mm^2$ form factor, a pin header type connector, and is electrically identical to the PC's ISA bus. The PC/104-*Plus* additionally provides a PCI bus, but so far only 10–20%[4] of PC/104 modules do. For some, or even most, applications, ISA would just provide sufficient throughput, but whether its deployment at this point in time is sensible is questionable. The board size of PC/104 may turn out to be a limiting factor.

Both CompactPCI and PC/104 are suitable for building a compact arrangement of circuit boards which can be placed in a shielded enclosure, together with a battery if desired. The sealed plastic cases known as "Pelican cases"[5] should be considered.

[3] http://www.pcisig.com/
[4] http://www.pc104.org/faq/
[5] http://www.pelican.com/

When designing equipment, the number of available slots for a bus system is one of the factors that must be considered when deciding about its deployment. CompactPCI offers 8 slots, with additional sets of 8 with the use of bridging circuitry. 3 slots would be taken up by the CPU, GPS, and optical Ethernet cards; the remaining ones are available for ADC cards. Assuming 3 ADC channels per card, a total of 15 channels per front-end can be implemented. This may provide more channels than are practically useful. Any fewer than 3 ADC channels per card is likely not to be cost-effective.

This is a further advantage of CompactPCI over standard PC boards, where the number of PCI slots tends to be limited to 5. With PC/104, the maximum number of cards which can be stacked up would only be limited by the maximum physical length of the bus signals.

11.4.2 System Configuration Issues

If more than one front-end ADC system is needed at the same location, the same satellite time source can be used to provide synchronisation. This becomes more economical with a larger number of channels per front-end. It is not as straightforward as having only one time source in the base station. However, the need for very time-critical transfer of ADC clock signals from the base station to the front-end disappears. With continually decreasing cost of GPS receivers, the possibility of providing each front-end with its own GPS receiver becomes a viable alternative.

Data needs to be transferred from the front-end to the base station by some kind of network. The internet provides an array of hardware and protocols, the obvious choice being Internet Protocol (IP) over Ethernet. A 100M Ethernet link provides about 10 Mbyte/s throughput, which is more than sufficient for power quality applications. PCI cards with fibre-optic 100M Ethernet interface are readily available, and any operating system should be able to handle internet protocols. This is discussed further in section 11.5.

The USB (Universal Serial Bus) [222] and IEEE1394 high-speed serial buses [75–77] are commonly found in desktop PCs. They are primarily designed for connecting portable devices, most notably storage devices and digital cameras. Their bandwidth is several times higher than 100M (Fast) Ethernet, but their use would be restricted to the benchtop. They are not suitable for local area applications because of their maximum cable lengths of 5 m or less [168].

11.5 Networking and Time Transfer

An alternative arrangement would be to transfer raw data to the central PC for processing. The maximum theoretical bandwidth of 100M Ethernet is ≈10 Mbyte/s. If 6–8 channels produce a total data rate of about 600–800 kbyte/s, the raw data of several dozen channels can be transferred to a PC over the same Ethernet segment. This would decrease the numerical processing requirements of the front-ends, at the expense of increasing it for a relatively cheap desktop PC. Considering the low cost of dual-CPU PCs, this is certainly a viable alternative. If e.g. a 1 GHz Celeron provides sufficient computing power for a particular application, a whole computer can be purchased in a case the size of a CDROM drive. In case such a physically small computer is used, a 100M hub/switch will be required as well, which collects all the optical cables from the front-ends into a single twisted pair cable connected to the computer. Ethernet is the de-facto networking standard, and components for it are available abundantly and cheaply.

As discussed in section 11.3, it can be desirable to use a sampling frequency which is a multiple of the mains fundamental. This necessitates that the actual mains frequency is known by the front-end's sample clock generator, but the measurement of the mains frequency does not have to be performed by the front-end itself. To reduce the hardware required by the overall system, measurement of the mains frequency can be performed by the site-wide PC to which all the front-ends are connected. The frequency can then be transmitted via the Ethernet link, although there are limits to the accuracy of the time transfer which can be achieved with this. It is not necessary to do this every mains cycle because the mains frequency change per cycle is small. If a measurement

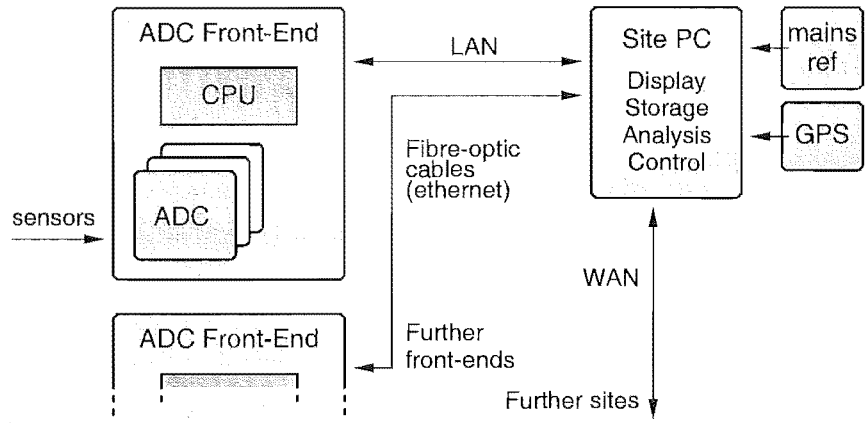


Figure 11.3: Another possible configuration of the proposed sampling hardware, utilising the LAN for time transfer. A 1 µs accuracy with this can only be achieved with exceptional effort.

site has more than one power distribution system, each system might run at a slightly different frequency. Multiple frequencies can be measured by the site-PC and communicated to the front-ends by using TCP broadcast messages. If for any reason a mains-synchronous signal is required on the front-ends because of their specific design, measuring the mains frequency on the site-PC does not offer any advantage.

The GPS receivers are also an area where costs can potentially be reduced. Instead of providing each front-end with a GPS receiver, a single receiver can be connected to the PC common to a measurement site, as shown in figure 11.3. A time-transfer must then be implemented from the PC to the front-ends. An obvious candidate for this is the network time protocol NTP [157–161], which was designed to perform this task over a network with unknown and changeable delays. It compensates for the timing characteristics of the network as much as possible. Each NTP node by default acts as both server and client, but in the configuration of figure 11.3 the server would be on the PC, with a client on each of the front-ends.

The signal propagation delays are not negligible. The lower bound is the speed of light, at 334 ns/100 m. This delay can be ignored if all sites use the same cable lengths with all front-ends. Alternatively, the actual delay can be compensated for. Building the system and the software with the ability to compensate for this delay is essentially necessary anyway: when relating data from different sites, the samples will have to be matched by their time stamps with some closest-fit method. In practice, the delays of packets travelling over the Ethernet link exceeds the signal propagation time by one to three orders of magnitude.

To reduce overhead and delays, NTP uses the UDP/IP instead of TCP/IP protocol. UDP does not have a connection state. The difference is in the number of packets required. For obtaining the time once:

TCP, 8 packets:	UDP, 2 packets:
→ open	→ request time
← ack	← receive time
→ request time	
← ack	
← receive time	
→ ack	
→ close	
← ack	

Even if the TCP connection is kept permanently open, twice as many packets would be required.

The characteristic of the Ethernet frame and IP packet handling largely determines the accuracy of the time transfer. Quality of service features of IP can be used to assign priorities to certain packets, and tp mark the NTP packets with a higher priority. This reduces latency by moving

them to the head of the transmit queue, and therefore increases the accuracy of the time transfer. This can for example be achieved with Linux by setting the respective routing options (package iproute2). Any intermediate network devices however would also have to forward these packets at the highest priority. Simple switching hubs (switches) are not capable of doing this, which effectively necessitates to equip the site-PC with a sufficient number of optical ports to connect all the front-ends directly. Alternatively, more expensive switches can be used.

In principle, delays over Ethernet come from the time it takes for packets to enter the transmit queue, how long they stay in the queue, how long it takes to transmit them once they have reached the front of the queue, and time time spent in the receive buffer. The time it takes for a packet to go over the wire is proportional to its size, and collisions on the physical link can cause delays. The timing of the interrupts for handling the IP protocol and the network card also have a large effect on overall network latency.

The basic network diagnostic tool "ping" can be used to obtain an indication of packet transfer speed. Ping sends off a small packet with the meaning of "hi, are you there?", expects a response packet saying "yes", and outputs the round-trip delay. (The absence of a "yes" response is in itself a valuable diagnostic answer.)

The round-trip time for ping packets between the aforementioned P3/Duron is ≈250 µs. The time for two Opteron computers linked via a switch is ≈150 µs. Soliciting a response from the local computer itself cuts the physical link and network card drivers out of the loop. This can be tested by pinging either the local computer's own external network interface, or the localhost interface. The delays of a P3-450 are ≈130 µs, and of an Opteron 244 (1800 MHz) under full CPU load ≈40 µs. There is no doubt that a faster CPU reduces the Ethernet latency.

To assess the behaviour of NTP over a typical network, a number of NTP installations in active use were queried for their own assessment of accuracy and the networks' timing characteristics. The output which was produced by querying the NTP software is given in figure 11.4 and shows, among other things, the network delay, the difference (offset) of the remote and local clocks, and the jitter of the network communications, all in milliseconds, and the estimated error for the local clock. Unless otherwise noted, the software in use was SuSE Linux 8.2 with xntp 4.1.1 for both server and client.

Although the NTP jitter is sometimes reported to be as low as 15 µs, it should not be seen as a typically achievable value. Low jitter is only reported if there is no other network traffic, which is unrealistic. With moderate network traffic, the jitter seems to be typically 0.5 – 1 ms. The accuracy which can be achieved decreases with jitter, and after some delay also with CPU load, which is not surprising. The first of the above 3 test cases shows that ≈1 ms accuracy can be achieved over a multi-segment LAN without special effort. A cable-modem link to the ISP's time server still provides an accuracy better than 1.5 ms. Connecting two computers directly was expected to result in a noticeably improved accuracy for the time transfer because the indeterminate delays of the intermediate network infrastructure devices are absent. However, the results are disappointing: as the third case above shows, the achievable accuracy is in the same order as for a LAN.

Alternative algorithms to NTP exist [120, 121]. Perhaps they are more suitable, but NTP implementations are readily available and there seems to be little difference in performance: an uncertainty of 2 ms over 1200 km is easily achievable [120] with NTP. The situation for power quality monitoring is different: most facilities are smaller than 100 m, and the network topology is relatively simple. Both of these factors negatively affect NTP accuracy.

An accuracy of NTP over long distances in the order of 1 – 10 ms seems to be generally accepted. Caporaloni [23] found a typical rms value of 35 ms and that this value *"looks reasonable when compared to the intrinsic limits of the Windows operating system"*. The processor speed of the hardware used is not given. Schmid [194] claims an average accuracy of 10 ms. NTP has benefited from speed improvements of computers and network over time. On LANs, reliable synchronisation to less than 1 ms has been demonstrated [161]. This however is three orders of magnitude away from less than 1 µs.

Considerable improvement is possible by optimising the Ethernet transport software, and linking it more closely with the NTP software. The basic aim is to prioritise the communication between

Setup: The NTP server on host cantva is running under VMS on a DEC computer. Host pukeko is a Unix machine. Both servers are reached from the client via a number of switches, some only 10M. All on the university's LAN.
Result: offset 1.2 ms/750 µs, delay 3 ms, jitter 15/100 µs, estimated error 700 µs

```
ppp2|root[1]:~# ntpq -c peers
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+cantva.canterbu pukeko.cosc.can  2 u   29   64  377    3.147    1.265   0.015
*pukeko.cosc.can .GPS.            1 u   36   64  377    2.997    0.746   0.109

ppp2|root[1]:~# ntptime
ntp_gettime() returns code 0 (OK)
  time c3d18f6b.62fa5000  Mon, Feb  9 2004 17:47:39.386, (.386632),
  maximum error 96523 us, estimated error 688 us
ntp_adjtime() returns code 0 (OK)
  modes 0x0 (),
  offset 640.000 us, frequency -69.184 ppm, interval 4 s,
  maximum error 96523 us, estimated error 688 us,
  status 0x1 (PLL),
  time constant 2, precision 1.000 us, tolerance 512 ppm,
  pps frequency 0.000 ppm, stability 512.000 ppm, jitter 200.000 us,
  intervals 0, jitter exceeded 0, stability exceeded 0, errors 0.
```

Setup: Using the ISP's time server via a 128 kbit cable internet connection. The time server is possibly not in the same city, and the software running on it is unknown.
Result: offset 600 µs, delay 14.5 ms, jitter 200 µs, estimated error 1.2 ms

```
hihi|root[1]:~# ntpq -c peers
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*dns2.paradise.n ntp1.cs.mu.OZ.A  2 u  423 1024  377   14.482    0.585   0.201

hihi|root[1]:~# ntptime
ntp_gettime() returns code 0 (OK)
  time c3d194c4.b6cde000  Mon, Feb  9 2004 18:10:28.714, (.714079),
  maximum error 314018 us, estimated error 1172 us
ntp_adjtime() returns code 0 (OK)
  modes 0x0 (),
  offset 525.000 us, frequency -93.126 ppm, interval 4 s,
  maximum error 314018 us, estimated error 1172 us,
  status 0x1 (PLL),
  time constant 6, precision 1.000 us, tolerance 512 ppm,
  pps frequency 0.000 ppm, stability 512.000 ppm, jitter 200.000 us,
  intervals 0, jitter exceeded 0, stability exceeded 0, errors 0.
```

Setup: Two computers connected via 100M Ethernet directly with each other. The NTP "server" (hihi) was synchronised to itself, with jitter 8 µs, estimated error 10 µs.
Result: offset 3.2 ms, delay 500 µs, jitter 150 µs, estimated error 800 µs.

```
ruru|root[1]:~# ntpq -c peers
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*hihi.local      LOCAL(0)        11 u   60  128  377    0.505   -3.241   0.136

ruru|root[1]:~# ntptime
ntp_gettime() returns code 0 (OK)
  time c3d19a7d.20283000  Mon, Feb  9 2004 18:34:53.125, (.125613),
  maximum error 53690 us, estimated error 836 us
ntp_adjtime() returns code 0 (OK)
  modes 0x0 (),
  offset -2876.000 us, frequency 27.861 ppm, interval 4 s,
  maximum error 53690 us, estimated error 836 us,
  status 0x1 (PLL),
  time constant 3, precision 1.000 us, tolerance 512 ppm,
  pps frequency 0.000 ppm, stability 512.000 ppm, jitter 200.000 us,
  intervals 0, jitter exceeded 0, stability exceeded 0, errors 0.
```

Figure 11.4: NTP peer timing.

the NTP client and server. This can be achieved by a number of means to a varying degree. On a multi-tasking system, process-priority for the NTP software can be increased. Network packets associated with NTP can be given higher priority, by tweaking the packet handling code's parameters and/or by using quality-of-service features. The largest gain can be achieved by moving the time-critical part of the NTP algorithm, the time stamping of sent and received requests, to a high-priority interrupt and possibly even away from the direct control of the operating system. This should not be too difficult. On a real-time OS for embedded systems pretty much all the software is tailor-made. On a general-purpose OS and PC-like hardware a spare interrupt should be available as well, but the time-stamping modifications need to be applied to the low-level Ethernet code. On a configuration with general-purpose OS positioned on top of a real-time base, time-stamping could be easier to implement.

Most of these methods have been implemented by developers at ABB [202]. Depending on the level of accuracy desired [64], software and a high-priority interrupt are sufficient. The highest accuracy can only be achieved with specialised hardware on both the client and server, but 1 µs is a possibility.

An accuracy of ±25 µs has been achieved by using hardware time-stamping on the NTP server, and an Ethernet hardware driver on the highest-priority interrupt running independently of the real-time OS [202]. The hardware on the client side was off-the-shelf Ethernet. The NTP server used was integrated into a commercially available network switch which implemented time-stamping of packets in hardware and was equipped with a low-drift oscillator. To reach 1 µs accuracy, time-stamping of packets on the client must also be performed in hardware. No other network devices were permitted between client and server in either case. The network was running at 100 Mbit/s (Fast Ethernet), using the standard SNTP protocol [157].

NTP is worth considering for power systems instrumentation, but its accuracy is limited with low-cost implementations. The decision should be made on required accuracy and available budget. An accuracy of ±1 ms (IEC 61850 class T1) can easily be achieved with off-the-shelf software and hardware over a LAN. A ±25 µs accuracy (class T3) would require a specialised time server, which can be integrated into a network router. An accuracy of ±1 µs (class T5) requires specialised hardware and software for both server and client, and a restricted network topology. Although the network topology restrictions are mostly inconsequential in power systems practice, the cost of the componentry has to be weighed against the cost of multiple satellite time receivers.

11.6 System Configurations and Third-Party Hardware

In the previous sections of this chapter, advances in processor and network technology were discussed, and possible structures of a new data acquisition system for power quality monitoring outlined. In this section, four different system configurations are outlined with varying trade-offs between performance and cost. Some commercially available hardware which could be used to build these systems is given; this is not meant to be an exhaustive presentation, only an indication of what is possible and readily available.

11.6.1 PC With Standard A/D Converter Card

By far the easiest and cheapest solution is to use a standard PC, commercial ADC cards, and a GPS receiver. The software supplied with the ADC card is probably useful, but further programming for time stamping and for interfacing to GPS is needed. Time stamping can be performed while the data is being read from the ADC card(s). An accuracy of 1 µs can not be achieved, but low milliseconds should be easy even with a general-purpose operating system. Using a real-time OS would improve the accuracy somewhat, but software development for such a system carries considerable cost. A hybrid real-time/general OS would be a worth-while compromise and should achieve a better than 1 µs accuracy while maintaining the benefits of a general OS. Mains-synchronous sampling is possible as long as the ADC cards allow fine-tuning of the sampling frequency.

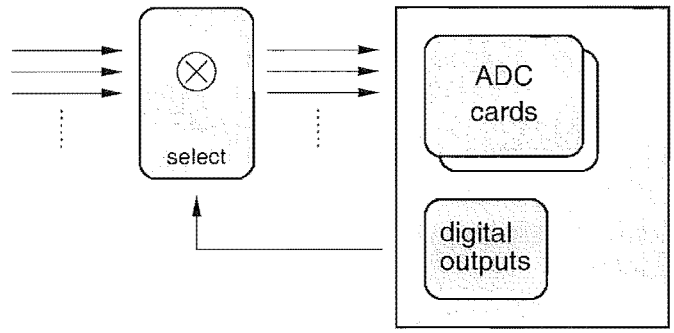


Figure 11.5: Using an external programmable pre-scaler and digital outputs to implement auto-ranging.

A setup like this would be suitable for deployment indoors, for example in a sub-station building or control room, where the signal sources are close enough to the PC to avoid signal distortion. A UPS (uninterruptible power supply) can be connected if desired. Connection to a LAN is straightforward. The possibility of WAN communications depends on the infrastructure available in the particular building.

11.6.2 Off-the-shelf Hardware

The general configuration of a system using off-the-shelf hardware is indicated in figure 11.2. The site-PC can be any PC with a general-purpose operating system. A large hard disk for storing collected data, and perhaps an option for archiving/backup, e.g. a DVD recorder, is an advantage. An Ethernet interface is now almost always included. For connecting the front-ends, optical Ethernet cards can be fitted, or a switch with optical connectors can be used instead.

For the front-end computer, a number of different bus systems are available. These should be evaluated in conjunction with the cards commercially available for them. Suitable contenders include CompactPCI, PCI, and PC/104.

A large number of x86 CPU boards, interface cards, a GPS card, and system racks for CompactPCI can be purchased from EKF[6] (EKF Elektronik GmbH). Using the GPS card, a tidy and compact system can be built in a relatively small volume. The cards do not appear to be low-cost, and ADC cards have to be sourced from another company. A selection of x86 and PowerPC CPU boards is available from MEN[7] (Mikro Elektronic GmbH Nürnberg); all these cards run at least Linux and VxWorks and several also run OS-9, QNX, and MS-Windows.

The ideal ADC card has 3–6 simultaneously sampled inputs, and a finely programmable sampling frequency. A sampling-clock PLL which can synchronise with an external signal can be used to achieve mains-synchronised sampling, although it would be simpler to program the ADC card with a sampling frequency derived from a mains frequency which is measured elsewhere. The sampling clock is connected to a high-priority interrupt on the CPU for time-stamping. ADC cards where the relation between a particular sampling clock pulse and the corresponding sample datum can not be established (e.g. delta-sigma converters, or those with FIFOs) are not suitable. Whether auto-ranging inputs are possible depends on the ADC cards featuring programmable input scalers.

Often a number of simple digital output lines can be found, e.g. from a parallel port or on a CPU or peripherals card. These can be connected to a separate PCB which contains a programmable pre-scaler, as shown in figure 11.5. Any desirable analog filters can be placed on this board too. A place to mount this circuit board can easily be found. It removes one requirement from the ADC card and thus increases the probability of finding a suitable one commercially.

WBC offers the PPChameleon[8] CPU module and a board with 3 PCI slots and several peripherals

[6] `http://www.ekf.de/`
[7] `http://www.men.de/`
[8] `http://www.wbc-europe.com/hot/news/company/ppc_modul.phtml`

including Ethernet and RS-232C. The CPU module includes dynamic and flash memory. With the board this makes up a complete PC with very low power consumption. Although the board is meant as a development system for hardware using the CPU module, there is no reason not to use both together as a final system. A complete Linux development environment and a ready-to-boot Linux system with real-time core are included with the board. The 3 PCI slots may become a limiting factor.

MEN similarly has a 400 MHz MPC8245 PowerPC CPU module[9] and a starter kit[10] with IDE, floppy, VGA, Ethernet etc. interfaces.

If an Ethernet port with an RJ-45 connector is available as part of e.g. a CPU board, as with the PPChameleon, a media converter can be used instead of having a slot taken up by a card with an optical port. The Allied Telesyn AT-LMC100SC and AT-LMC100ST[11] convert RJ-45 to 100FX with SC or ST connector. The maximum distance for the optical cable is 2 km.

A small form factor PC should be considered where a high number of slots are not required. For example, the Shuttle SN85G4 is $20 \times 30 \times 18\,\mathrm{cm}^3$ and weighs 2.85 kg with an aluminium case. PCs with ITX form factor are commonly available. These designs usually have no more than one PCI slot. Additional shielding for deployment in noisy environments may have to be provided. An online market overview[12] is useful for orientation.

11.6.3 Custom-Built A/D Converter Card

If suitable ADC cards are not commercially available, or not at an acceptable price, a custom-designed card should be considered. Apart from the ADC converters themselves, a number of other parts are needed as well. The bus interface is best implemented with a commercially available chip. Several models are readily available for PCI and CompactPCI. Availability of drivers for this chip for the operating system to be used is of high importance to reduce development time and therefore cost, unless the development team includes programmers who are exceptionally skilled in programming hardware drivers for the required operating system.

The acoustics research group [3, 52] has programmed a driver for Linux for the PLX PCI9054 bus interface (PCI↔IOBus) chip from PLX Technology, Inc., and has agreed to make it available on request under a GPL license.

A variable frequency oscillator for the sampling clock is not necessarily an important feature. For reasons of coherency between channels, it is better to have only one clock source shared by all ADC cards and all channels. This clock can be generated on one of the ADC cards and distributed to the others, but generating the clock by the CPU might be more appropriate. It is important that the ADCs can be switched to an external clock source (which may come from the CPU). Mains-synchronised sampling can be achieved by programming an appropriate clock frequency. It is paramount that the circuitry is designed in such a way that the relationship between individual clock pulses of the sample clock, and the sample data that is read from the card afterwards, can be correctly established. Furthermore, it must also be possible to relate each sample clock pulse to the system's method of time keeping with an accuracy of better than what is required for the overall system. The higher silicon integration available today has not only produced much faster CPUs, it is now also possible to place most or all digital circuitry needed for an ADC card into e.g. a single FPGA.

If analog filters and input pre-scalers are deemed necessary, they should be placed on the ADC card if possible. It is necessary to find an appropriate balance between the maximum number of channels which fit on the card, the maximum card area available, and the higher CPU-time that is required to achieve the same amount of filtering, if more of the filtering is performed by the CPU instead of on the ADC card.

[9] http://www.men.de/download/pdf/datasheet/15em04-.pdf
[10] http://www.men.de/download/pdf/datasheet/08ek02-.pdf
[11] http://www.alliedtelesyn.com/product/?LMC100
[12] http://matrix.sfftech.com/

The maximum size of a 3U CompactPCI card is approximately Eurocard size ($100 \times 160\,mm^2$). In theory, a PCI card can extend to the full depth of a PC, but in practice this is not exploitable for a system with limited enclosure space. In practice, the usable size of a PCI card is also approximately Eurocard size. If designed as a PC/104 card, it is easy to stack a number of cards holding fewer channels each. However, the bus interface would have to be duplicated per card, which may make PC/104 less economic.

A very strong argument in favour of PCI is that the cards can also be used in standard PCs. This would be an excellent and cost-effective approach for any software development, especially if the operating systems used on the front-end and the development PC are more or less identical. This would be the case for Linux, Linux with a real-time base, and MS-Windows with a real-time base, but not for dedicated real-time operating systems from various embedded systems software companies like Wind River or Greenhills.

Being able to use the ADC cards in standard PCs also opens up the possibility of a number of different system configurations. When it is not necessary to have the front-ends separate from the site-PC as shown in figure 11.2, they can be combined into one unit and deployed in a similar manner to that discussed in section 11.6.1. The issues for interfacing time signal receivers are identical to those for separate front-ends and those for a system as in section 11.6.1.

Implementing the ADC card in CompactPCI instead of PCI form factor opens up more possibilities for deployment in front-end systems as shown in figure 11.2 or figure 11.3. The disadvantage is that using them in a standard PC requires a suitable adapter. However, there is more space available in a standard PC for such an adapter.

11.6.4 Custom-Built Hardware

Designing every piece of hardware in-house gives the greatest flexibility for the design of the time stamping and sampling circuitries, and the largest possible freedom for selection of each individual component. The design of these parts for CHART III can be copied more or less as is, or be improved. All the circuitry fits into a contemporary FPGA.

A mostly custom-build front-end system offers the highest integration density, although it is not necessarily much higher than that of a CompactPCI system. Designing a GPS receiver in-house is unlikely to ever be cost-effective. Receiver modules with a serial interface and 1 PPS output packed on a small circuit board, as used by e.g. EKF[6,13], should be bought in. It is also not likely to be economic to re-invent the wheel for CPU cards or modules. CPU modules like the one from WBC[8] or MEN[9] already have memory and a bus interface included.

Designing the ADC card component of the system, as outlined in section 11.6.3, should be considered for the possibility of deployment in different situations. The ADC design issues discussed in section 11.6.3 also apply for a full custom design.

Custom-designing and manufacturing any hardware is expensive in both development time of qualified engineers and development equipment such as logic analysers or in-circuit emulators. It is only justifiable either if a system which meets the specifications can not be assembled from commercially available parts, or if a high number of systems is to be produced. For high production numbers, ASICs (application specific ICs) are cheaper than FPGAs.

11.6.5 Other Hardware of Interest

This section introduces a number of hardware items that may be of interest for some designs.

The briQ[14], a network computer from Terra Soft with the same form factor as a CDROM drive, is based on a 500 MHz G4 PowerPC and includes hard disk, memory, Ethernet, RS-232C and a small

[13] `http://www.ekf.de/c/cgps/cgl/cglpie.pdf`
[14] `http://www.terrasoftsolutions.com/products/briQ/`

display. It can provide a PCI slot with the use of an adapter. It can be used for remote data processing anywhere on a network, but does not have any graphics hardware and therefore one can not directly connect a monitor. It runs Linux without modifications. At US$1500, it is not cheap.

For approximately the size of a CDROM drive, the NanoII[15] is a CDROM drive, and also has an integrated 1.2 GHz Celeron (or others), 128 MB RAM, hard disk, and USB, parallel, serial, keyboard, mouse, audio, VGA and Ethernet interfaces. It comes with Linux pre-installed. Useful whenever computing power is wanted in a small case, for portable or fixed applications. Should be compared against a laptop.

The net4801[16] from Soekris is a single-board computer with 586-class CPU, up to 256 MB SDRAM, CompactFlash adapter, IDE interface for 2.5" disks, USB and serial interfaces, 12 bits general purpose I/O and 3 100M Ethernet ports. It is mainly meant as router/firewall, and runs BSD, Linux, and most real-time operating systems.

A list of small single-board computers is at LinuxDevices.com[17]. The same site also has a vast amount of information about embedded systems hardware, software, and training material.

11.7 Software Considerations

11.7.1 Operating Systems

The choice of software for the site-PC computer (see figure 11.2) is reasonably straightforward. The sample data is already time-stamped by the ADC front-end when it arrives for processing. None of the tasks are time-critical in the sense of real-time, and communications are via Ethernet, which in itself provides a certain amount of buffering. Desktop CPUs provide ample power for number crunching, with slight differences between manufacturers (see section 11.2). Multi-CPU systems, or symmetric multi-processor (SMP) systems, are available and cost-effective. A general-purpose operating system is well suited. Linux has proven to be reliable and would be a good choice for a dedicated instrumentation system. In particular, it enables users to log in remotely, which allows full use and control of the instrumentation system and its software without having to be on site.

Using a general-purpose OS enables the sharing of data analysis and visualisation software between the site-PC and any other PC which carries a copy of the sample data or pre-processed sample data, and makes data archival on e.g. CD, DVD, or tape trivial.

For the ADC front-end, a real-time OS like VxWorks or an equivalent from another supplier could be used. The drawback is that although these systems guarantee a response within a certain time, their average throughput can be rather low. When resources (other than CPU time) are depleted, they simply stop working altogether. This is the case with the iRMX real-time OS — when no free memory is left, it crashes. It is made worse by the fact that there appears to be a memory leak (some memory is not de-allocated after use) in a part of the OS itself, which means that an eventual system crash is inevitable. It should be noted that iRMX is no longer on the market, and that contemporary commercial real-time OSes are probably better designed.

In contrast, general purpose operating systems continue when resources are depleted, but at a much lower speed. In the case of memory exhaustion, this is made possible by configuring virtual memory (swap space). Under these conditions the instrument can no longer perform the intended task adequately, but the OS still allows the user to log in and fix the problem. This is an important consideration for an autonomous system which can be operated remotely.

For a data acquisition system, the real-time requirements are rather simple. With an external signal, counter values or ADC output values need to be saved within a short time frame. With

[15] http://www.linux-works.com/html/nano2.html

[16] http://www.soekris.com/net4801.htm

[17] http://www.linuxdevices.com/articles/AT8498487406.html

a 5 kHz sampling rate (50th harmonic at 50 Hz), ADC data of each channel must be saved every 400 µs. This interval is too short for regular process scheduling and therefore this task has to be performed in an ISR. Sampling rates much higher than this necessitate the use of DMA (direct memory access) or a CPU with sufficiently low interrupt latency. Even with higher sampling rates, it is not impossible to run a general-purpose OS.

A general-purpose OS can be run on a real-time kernel [172], but this is more sophisticated than is required. A sufficient solution is to modify the Linux kernel and to provide a fast interrupt routine. The availability of the source code makes this possible, and versions of Linux for embedded systems already exist. Considering that a reasonably sized Linux system easily fits into currently available flash memory modules and SDRAM sizes, there is nothing to be gained by using a Linux for embedded systems instead of the regular one, unless the design is for a large number of front-ends.

Software for storing the programs, and exchanging control information, programs and data with the base system, is already taken care of by a general-purpose OS, simplifying that part of the design considerably.

The fastest execution of the ISR which handles the time stamping can be achieved by permanently reserving one or two CPU registers for use by this ISR only. Reserving the registers saves the time of saving their contents on entry to the ISR and restoring them on exit from the ISR. They can be treated as read-only by the remaining code. Short of writing all the code in assembler (which is impossible for a project of this scope) the compiler has to be told not to generate code which makes use of the reserved registers. With the GNU C compiler[18] gcc [47] it would be possible to modify the target CPU's machine description and reduce the number of available registers, and then to recompile the compiler. A condition for this is that the CPU has a reasonably large number of identical registers, which is true e.g. for the PowerPC. As the remaining code has to run in fewer registers, a slight performance penalty has to be expected.

For a non-centralised data processing configuration, processing is performed by the ADC front-end itself, which requires more CPU power and leaves the central system to deal with storage, display and archiving.

11.7.2 Development Environments

A significant aspect for consideration is the choice of development tool for the system, especially for the front-end. Software for the site-PC can be developed with whichever development tools are usually used for software running on the operating system used for the site-PC.

Numerous operating systems are on the market for embedded systems, either real-time, or general purpose with and without real-time extensions. The latter category includes Linux and MS-Windows. These are typically sold with an integrated development system, and the development systems often run on a range of host systems and offer a range of target systems. Some companies require royalty payments for each copy of their embedded software in use whereas others do not.

There are advantages in using the same OS on the front-end as on the site-PC, if possible. If so, much of the software can be developed and tested on any PC, especially if the ADC cards used at the front-end can also be plugged into a PC.

If a real-time OS is used on the front-end, the companies selling real-time operating systems also sell suitable development systems with them. This includes Linux with real-time extensions.

For example, Ecrin[19] sells a number of boards and a development environment for Linux on the target system, available for Linux and MS-Windows host operating systems. Ecrin offers development of custom hardware and software on a commercial basis as well.

[18] `http://gcc.gnu.org/`
[19] `http://www.ecrin.com/uk/`

Green Hills Software[20] offer three different real-time operating systems, including Linux, all royalty free, and an extensive development environment[21] with a number of programming languages which runs on Linux, MS-Windows, Solaris and HP-UX. Green Hills claims to have the fastest compiler for PowerPC code. An industry source states that a simple but representative speed test shows a negligible difference to gcc.

Lauterbach Datentechnik GmbH[22] offers a range of tools like in-circuit debuggers and emulators for an impressive range of host operating systems and target CPUs.

Wind River Systems[23] have the largest market share with their non-royalty-free VxWorks, and also offer development tools and associated services. They have traditionally not supported Linux, but seem to now be considering it commercially worth their while[24].

11.7.3 Programming Languages and Application Software

Scripting languages promise rapid prototyping at a high level. They can also be used to extend applications to provide users with an easy means to interface short user-programs [14], e.g. for data analysis, to the control application. An example of using Python for message passing (i.e. data exchange) exists [16]. Many of these issues apply to the scenario of controlling multiple power quality measurement systems from a common station.

Kale compares various scripting and other programming languages [92]. Scripting languages have a reputation for introducing large computing overheads and being slow. While this is almost certainly true for some, Python seems to show that this need not be the case [178].

11.7.4 Analysis Software

A general-purpose instrumentation system should be as flexible as possible in the analysis software which runs on the front-end (or on the site-PC, if a system configuration as in figure 11.3 is implemented).

A user-selectable set of analysis software can be provided as a number of building blocks. Modules for common applications should be built into the system, and flexibility can be provided by allowing for easy inclusion of user-designed modules. Both built-in and user-designed modules are implemented as shared libraries with a defined function API (application program interface). On a general-purpose OS, shared libraries are commonly used, and can be loaded by the main program any time after startup. This is a convenient technique for loading program modules during the runtime of the main program. Each shared library comprises one building block. Building blocks may also be called plug-ins and can include:

- reading sample data
- FFT
- transient detection
- saving critical values which exceeded a threshold
- generating alerts when pre-set conditions are met

[20] http://www.ghs.com/
[21] http://www.ghs.com/products/MULTI_IDE.html
[22] http://www.lauterbach.com/mindex.html
[23] http://windriver.com/
[24] http://linuxdevices.com/news/NS6468351784.html

An alternative to an implementation as shared libraries is an implementation as stand-alone program in the style of a classic filter: reading input from standard input and writing output to standard output.

Each plug-in has an associated "cost" of CPU and memory bandwidth use. The total cost of all building blocks combined by the user must stay below 100%, with 100% being the system performance left over after the requirements for the time stamping and the operating systems are deducted from the total provided by the hardware.

Further possible analysis functions, which can also be implemented as plug-ins, include:

- Capturing and analysing harmonics that exceed certain boundaries, or are outside certain shapes. Also capturing to either side to show buildup and the dynamic reaction of the grid.
- Calculating power quality phenomena (including flicker according to published standards [59-62, 67], and logging and generating an alert when agreed or preset thresholds are exceeded.
- Recording details when line voltages are outside their nominal range.
- Recording details of any transients, including voltages and currents for a short period before and after the event.
- Keeping a continuous log of line frequency and transferred power.
- Keeping a permanent record of those grid parameters which were pre-selected by the user.

If this instrumentation system is to be connected to the internet, security considerations need to be taken very seriously. This becomes paramount if the front-ends are connected via the site-PC and the internet to a central PC. The encryption which will need to be used for this incurs a non-negligible CPU load penalty, which must be considered and the site-PC dimensioned accordingly. Experience has shown repeatedly that internet security issues have not received appropriate attention.

11.8 Commercial Data Acquisition Systems

Several complete commercially available data acquisition systems are examined, and some of their key features discussed with respect to the requirements established in chapter 4. This gives an indication of the level of performance which is currently available on the market.

These key features are time stamping accuracy, real-time processing of data, deployment location issues, and a sufficient number of channels. The minimum time stamping accuracy depends on the intended application, but the focus is towards 1 µs. Processing data in real time allows for continuous operation; instruments allowing only a few seconds of high-sample-rate recording are not considered. The deployment location is limited by the instrument's signal, environmental, and power supply specifications. The minimum number of channels is considered to be 3.

Of the well-known instrumentation companies, Philips, HP and Tektronix do not have any data acquisition products at all. The systems of Fluke and Dranetz-BMI do not have any time stamping features. Because time stamping is an integral aspect of power systems instrumentation as discussed in this thesis, instruments which do not have a minimum level of time stamping capabilities are not considered.

The HSDAS 2020 from BBT Products[25] is a 16 bit data acquisition system with 64 channels which can sample simultaneously in the MHz range. Sustainable data throughput to hot-swappable hard disks is in excess of 100 Mbyte/s. The system includes an optional GPS card for time stamping data and events. The details and accuracy of the time stamping are unclear, as is the channel coherency, and whether data can be accessed from storage while the system is still running. The

[25] http://www.bbtechno.com/prod.htm

DSAS 2010 system from the same company has a nanosecond channel coherency, but no time stamping feature. Both systems use a PC for data processing, and their deployment restrictions are the same as those of a desktop PC.

The ADS2-SIB product family[26] (previously ADS-3000) from Tech S.A.T GmbH and /dev Software GmbH is a data acquisition environment for "distributed real-time systems", is equipped with an impressive range of hardware and software features. Numerous industry standards are supported. Time synchronisation uses IRIG-B, can be equipped with a GPS time source and is accurate to typically ±1 µs. This system is marketed as an analysis tool for high-end applications such as avionics testing. It is implemented around the VME bus and would appear to meet all power quality instrumentation requirements. However, it is unlikely to be a low-cost solution.

Schneider Electric's powerlogic[27] series is a range of monitoring instruments for fixed rack installations. The top-of-the-line model CM4000T is equipped with pre-defined functionality for power metering, basic power quality monitoring, and transient recording. The event recording operates with a time stamp resolution of 1 ms. Time can be supplied externally via a GPS receiver. The practically usable accuracy is not better than the available resolution. This instrument is targeted at power metering.

Symmetric Research offers a data acquisition system[28] which connects to a PC's parallel port. The programmable sampling rate can be up to 5 kHz, and samples are time-stamped with an accuracy of 10 µs (typical). Full source code for drivers and control software is available for MS-Windows and Linux. Data processing is performed by the PC. The 4-channel version costs US$550, the 8-channel version US$950, and the GPS module US$340. This system doesn't meet the most stringent requirements, but will be sufficient for many tasks.

National Instruments offers a large range of modular data acquisition equipment. Sample data can be time-stamped with a software solution, but this will only give millisecond accuracy. A direct GPS-based time-stamping solution is not available, but the article *"Synchronizing and Correlating Measurements to a Global Timebase with GPS"*[29] describes how to connect a 32 bit timer/counter module (PXI-6608) with a DAQ module (PXI-6070) to time-stamp an analog signal. It is unclear how this scales to multiple signals, and what the channel coherency is. The error of the synchronisation of the timer/counter to a GPS receiver is ±300 ns.

The Hathaway IDM Data Acquisition System[30] features 10–32 analog inputs, a 16 bit ADC, event inputs, and basic power parameters calculations like rms, harmonics, and phasors. A GPS unit and a permanent storage unit can be attached. Multiple units can be connected together via Ethernet, which also connects them to a master controller PC. Although time stamping of data is mentioned, no accuracy is specified for the time stamping or the phasor values. Multiple units connected together are synchronised to within ±10 µs of each other. The units have 19" cases for rack mounting.

The Arbiter Systems Model 1133A[31] is marketed as a revenue meter and power quality monitor suitable for a single 3-phase system. Limits can be set on measured quantities. Data recording is limited, measured values can be recorded at most once per minute, or when an event occurs. Raw data can not be recorded or downloaded from the instrument. Recorded values are time-stamped, but the accuracy is not given. The time base is GPS-based and specified with an accuracy better than 1 µs (GPS locked). The system time accuracy is 1 µs plus time base error, the event input accuracy is ±10 µs (typical). The accuracy of phase measurements is only specified for the fundamental. The unit is in a 19" rackmount case.

The Rochester / Ametek TR-2000 Series Multi-Function Recorder[32] is available in a number of different configurations with up to 32 analog inputs, and user-programmable sampling rates of

[26] http://www.techsat.com/page.php?id=101
[27] http://www.powerlogic.com/
[28] http://www.symres.com/products/seldaq.htm
[29] http://zone.ni.com/devzone/conceptd.nsf/webmain/2FAD3B4E67BA766186256B9C00679F75
[30] http://www.qualitrolcorp.com/
[31] http://www.arbiter.com/ftp/datasheets/ds_1133a.pdf
[32] http://preview.ametek.com/content-manager/files/pip/tr2000specs.pdf

either up to 384 samples per cycle or up to 195 kHz sampling rate for transient recording. Channel coherence is unclear, but time stamping is specified to be accurate to ±100 ns, making the question of coherence practically irrelevant. Disturbances can be recorded at 0.5 samples/cycle, and power quality and phasor measurement functions are provided. Unfortunately the input voltage range is only suitable for American mains voltages. The unit is in a 19" rackmount case with internal GPS receiver. This unit would not meet all of the requirements outlined in chapters 4 and 5, and does not address the issue of the locations of signal sources, but it is the only one found on the market which meets the accuracy requirements for harmonic phase comparisons.

In summary, most instruments on the market are aimed at single-point measurements for the purposes of metering and power quality recording, but are not geared for distributed measurements. Instruments for fixed installation provide interfaces through which data can be obtained, and can thus be integrated into a larger instrument configuration. Although a GPS receiver is a common feature, it is not used for highly accurate time stamping of measured values. Only one instrument provides a time stamping accuracy of 1 µs or better and is therefore suitable for phase comparisons of higher order harmonics.

11.9 Conclusion

This chapter has shown that a modern power quality monitoring system can now be built much more simply and cheaply. A single modern CPU can now be used for continuous synchronous measurements of voltage and current vectors in power distribution systems, instead of expensive hardware-based solutions. In particular:

- Improvements in CPU performance mean that DSPs are no longer required. The improvements in performance have been associated with a lesser improvement in latency, but latency is now low enough for most, if not all, power quality measurements.
- FFT benchmarks indicated that a 32-bit Athlon, P4 or PowerPC processor is adequate to handle up to 48 data channels, plus data storage, network communications, sampling clock generation and time stamping.
- A 32-bit counter register within the CPU can be used for time stamping and to control the sampling frequency. With tight integration of hardware and software, a 1 µs accuracy can be achieved. With a standard operating system and current CPU, a 25 µs accuracy should be achievable with a high-priority interrupt.
- System cost has been reduced substantially: GPS unit price has been greatly reduced, fewer processors are required, and fewer specialised parts such as DSPs are required.
- Especially for applications which require time stamping of the order of 10–100 µs rather than 1 µs or less, it is possible to generate the time signal at a single processor and transfer it to each distributed front-end over an Ethernet LAN.
- A real-time operating system is not required, and general-purpose operating systems are simpler to work with. General-purpose operating systems based on a real-time core are a good compromise for applications which benefit from a limited amount of real-time functionality.
- Small volume systems can be designed, greatly increasing portability. For example, the electronics for 6–12 ADC channels can be fitted into a CompactPCI box approximately 100 × 160 × 80 mm^3, excluding power supply, battery, and shielding.

As described in this chapter, a number of different system configurations are possible, providing the required functionality for a range of applications. Such low-cost data acquisition systems would allow a greater use of informative quality monitoring of power systems.

chapter 12

Conclusions and Future Improvements

The work described in this thesis falls into three major areas, detailed in the following sections. First, the CHART III system was used in the field. Improvements were made as a result, especially to enhance its application as a data acquisition system for synchronised distributed measurements.

The next major aspect of the work of this thesis was a detailed investigation of the requirements for power quality monitoring systems, making use of the experience gained from the field tests. This lead to the generation of a template for defining a power quality instrumentation requirement specification. Recommendations for a suitable choice of analog-to-digital converters were verified with mathematical simulations of their error contribution.

Finally, the performance of currently available microprocessors was investigated, especially their actual speed in calculating Fourier transforms. Options for the design of power systems instrumentation using the current technology were then discussed. This part is completed with a brief overview of currently available commercial systems.

Four further papers have been prepared from the material in this thesis, and were submitted for publication. They are reproduced in appendix A (page 170 ff.), together with other publications which were prepared during the course of this work. The four papers are on the topics of sample rate and ADC width, system requirements, design considerations, and microprocessor advances.

12.1 System Requirements

The complexity of a power distribution monitoring system is most dependent on the number of channels and sites which must be analysed, and the required time stamping accuracy. If a single 3-phase system at a single site is to be investigated, only 3–8 channels are required, the ADCs can all fit into a single enclosure and synchronisation is easily achievable. At the other end of the complexity scale, if multiple 3-phase systems are to be investigated simultaneously at multiple sites, many tens of channels will be required and synchronisation requires in practice that all samples must be time-stamped. Synchronisation within each site (for example, by distribution of a time signal from a master clock over fibre-optic cables) and between sites (for example, by GPS) must both be implemented. For highly accurate time stamping, an off-the-shelf system can not currently be purchased, so design (or at least assembly from off-the-shelf components) is necessary.

Requirements are discussed in chapter 4 and listed in the form of a specification template in chapter 5. This template lists all the requirements which may need to be considered for different applications. In summary, other than sample synchronisation and time-stamping, the most

significant issues which must be considered in the design of a power quality monitoring system are:

- The duration of the measurements. Lightweight portable systems are best suited for short-term measurements, permanent installations may need to be more robust. Continuous operation requires a permanent power supply, and enough processing power and memory for pre-processing, processing, storage and transfer of data.
- Separation of the ADC front-ends from the control and main data processing unit (for example, in a control room) to locate them close to the signal sources for noise reduction. All parts of the system must be networked together. A more powerful and cost-effective PC can be used for data processing in the control room.
- Environmental protection (electromagnetic, weather)
- The input stage, including the required range of inputs, ADC width, sampling rate, oversampling and filters.
- Compensation for the characteristics of sensors used, and calibration of the instrument including the whole measurement process.
- Sampling modes, triggers and control of sampling.
- Data analysis algorithms, and determination and storage of data deemed to be of interest.

The combination of these diverse and technically challenging aspects leads to complex requirements for a system. Together with the complication of different applications focusing on different aspects, this complexity may well be the reason for the lack of such monitoring systems in the marketplace.

12.2 CHART III

CHART III is an example of a data acquisition system which fulfils these complex requirements for power quality monitoring, including the requirement for highly accurate time stamping. The system design and the time base (the DSM) used in CHART III were described in chapters 6 and 7.

The DSM hardware was designed by the author to be very flexible. The DSP has plenty of spare capacity, giving reserves for future applications. All essential functions like the real-time clock, event capturing, and the sample clock generator (SRM), are implemented using field programmable gate arrays (FPGA). All this can be molded into a completely different application without changing the hardware of the board. If this is not enough, the socketed gate array chip can be replaced easily by a larger capacity one. Because the FPGA configuration is uploaded by the DSP software, a different application would not require programming of any hardware.

A number of improvements were made to the time base to increase the versatility and ease of use of the instrument. Sampling controls were extended, and the sampling time, duration, and number of repetitions can now be specified to record e.g. some system variables once per minute. The time stamping now continues to operate with the stability of a crystal oscillator if the signal from the GPS receiver becomes unavailable. A number of time base error codes can be displayed on the front panel's LED in a coded form, giving a quick indication of the cause for a problem which may have occurred. All the status variables of the time base and the sampling can now be displayed on the control unit. The software of the time base was rearranged into the operating and application layer model used by the software on the other DSPs.

Two field tests were performed to gather experience with the use of the CHART III system for synchronised data acquisition. The first involved two of three ripple control signal injection plants in Christchurch. The power company was interested in measuring their phase offset to establish the load flow and their correct phase synchronisation. The test demonstrated the potential usefulness

of CHARTIII for this kind of measurement, and also highlighted a number of usability issues and the importance of a reliable clean power supply.

The second field test was performed at a ring feeder substation at the University of Canterbury. These tests established the limits of CHARTIII for data acquisition in the time domain while measuring the effect of an active harmonic compensator on the supply waveform to a net supplying a large number of PCs. Initial tests showed that the limits were far lower than expected, which was traced to a sub-optimal implementation of the data transfer from the DSP boards to the HUB. Modifying the data transfer to be asynchronous or interrupt-driven removed this bottle-neck, and it was found that at least 500 mains cycles (or 10 s) can be sampled reliably on 3–12 channels simultaneously before the system starts to malfunction because of the data overload.

For applications requiring many channels, major gains in usability could be achieved by producing software which could setup and control multiple channels at once. The program getsync was created for this. Furthermore, unlike the existing setup and control program, getsync is multi-platform and stable Linux or Solaris platforms can be used instead of the unstable MS-Windows 95. (The CADU program is made with a 16 bit machine model and will not run on newer versions of MS-Windows.)

The chartdat program was created for data analysis and offers a number of filtering and other functions. It is also multi-platform and capable of being used for batch processing. Gigabytes of data could (and can) be analysed on powerful Unix workstations instead of on the 80486-class machines current at the time. A detailed user manual was produced.

For real-time distributed measurements, the data from distant systems needs to be collected into one place for analysis. For this purpose, CHARTIII was networked via internet protocols over telephone modems. This also has the major advantage of making remote-control of CHARTIII systems possible. This setup requires one PC per CHARTIII system plus one PC at the central location to perform the necessary PPP, ARP and routing functions. To date, 80486-class machines with 8–16 MB of RAM are still sufficient for this task.

12.3 Error Simulations

The effects of quantisation noise on the ability to recover magnitude and phase information have been examined, and simulated with different parameters. It was found that

- Both magnitude and phase errors are independent of harmonic order.
- The magnitude error is independent of the magnitude of that harmonic, for magnitudes significantly larger than the noise magnitude.
- The rms phase error is inversely proportional to the harmonic magnitude.
- Both magnitude and phase rms error are directly proportional to the size of the quantisation interval, or the inverse of the number of quantisation steps.
- Both magnitude and phase rms error are inversely proportional to the square root of the FFT length.

Therefore, the precision of the instrument can be increased by increasing the ADC width or the transform length or both. The transform length is increased by increasing the sample rate. Depending on the application, it is likely that a 12 bit ADC is sufficient for power quality monitoring.

12.4 Advances in Technology

Power quality monitoring systems can now be built much more simply and cheaply then when CHARTIII was built, using current microprocessor technology. The most important improvement

is that a single standard CPU can now handle the data from a number of channels, eliminating the need for specialised digital signal processors and the associated cost of producing software for a second architecture. Processor performance seems to be set to increase steadily, promising future improvements of time stamping accuracy, the number of channels which can be handled by one processor, or more complex analysis functions.

As a representative function performed by virtually every power quality monitoring system, the computation of fast Fourier transforms was chosen as a benchmark to compare a number of different microprocessor architectures. The results indicate that a 32-bit Athlon, P4 or PowerPC processor is adequate to handle up to 48 data channels, plus data storage, network communications, sampling clock generation and time stamping.

Real-time operating systems are not needed for most applications. A general-purpose operating system has the advantages of being low-cost and offering straightforward software development. If soft real-time functionality is considered to be necessary, a general-purpose operating system placed on a real-time core should be investigated.

Time stamping can be performed by the CPU for applications that require less stringent time stamping, of the order of 10–100 µs. It is possible to generate the time signal at a single processor and transfer it to each distributed front-end over an Ethernet LAN. The network time protocol, NTP, is commonly used for this purpose.

It is common practice to use the GPS as an external time source, even for applications for which a reference to absolute time is not required, because using the GPS is more economical. GPS receivers are priced low enough that it becomes increasingly feasible to use a receiver in each enclosure instead of transporting the time between enclosures in close proximity. Instead of GPS, GLONASS can be used, or Galileo once it is operational.

For the most stringent accuracy classes, creating the time stamping solely with the help of a microprocessor requires tight control over the design's handling of timing. The use of NTP for time transfers is possible, but requires specialised hardware on both sides, reducing or negating any cost advantage.

A market survey of currently available commercial instruments has shown that power quality monitors are typically aimed at very low data rate single-location recording of power quality parameters. Only one instrument was found that is specified to handle sustained throughput with multi-site recordings at the tightest accuracy class.

12.5 Future Enhancements

The demand for power quality monitoring systems is likely to increase in the future as suitable systems become available at an acceptable price. Technological evolution in the area of microprocessors are the major contributing factor to a reduction in cost and complexity, which is expected to continue for the foreseeable future.

Further simulations could usefully be performed to include oversampling, which is expected to increase accuracy but was not investigated within the scope of the simulations undertaken so far. The effect of digital filters typically used in data acquisition systems could also be simulated. The use of some type of digital filter is required with oversampling to reduce the data rate by the oversampling factor.

The error simulations could be taken further to investigate the errors and effects resulting from sampling frequencies which are not a multiple of the actual mains frequency. Synchronising the sampling frequency to the mains frequency carries a cost in terms of hardware (e.g. zero crossing detection) and/or software (determining the mains frequency); eliminating this would make the hardware design simpler and free up processing resources for other tasks.

Much of the practical usefulness of a general purpose instrumentation system rests with the design of its software, and this should be carefully considered in any future work. The user interface more

closely resembles a computer program than e.g. an oscilloscope. The software for the control unit can be programmed as a desktop application for one or more of the contemporary operating systems. The software for the ADC units should allow basic network access to the system, for control of the unit and retrieval of data. When all systems are networked, analysis of data from multiple sites and archival storage can be centralised.

The CHART III system could be improved in a number of ways with the existing hardware and operating system. For example, further applications could be programmed and made available to users via the CADU or some equivalent program. Further sampling control would also be very useful. However, if the system were to be developed as a commercial instrument, it would be easier and more effective to redesign it using current technology. A superset of system requirements and a number of possible system configurations have been described in this thesis; the next step would be to determine the optimal feature set and configuration for the target applications. The experiences gained with the existing system would be of great benefit in this.

appendix A

Publications

A.1 Published

During the course of this thesis, the following items were published and are reproduced in this appendix:

- *"Synchronous Sampling with* CHART*"*

  Application note that describes how to perform synchronous measurements with CHART III. This note was supplied to CHART III customers. Reference [106], reproduced on page 153.

- *"Some Aspects of Precise Synchronisation of Data Acquisition for Power Systems Harmonic Analysis"*

  A paper discussing the requirements for monitoring harmonics in a power distribution system, and describing the CHART III system. This article was peer-reviewed and was presented at the Australasian Universities Power Engineering Conference (AUPEC) in Perth in September 2001. Reference [110], reproduced on page 161.

- *"Impact of Processor Evolution on Synchronous Measurements for Power Quality Monitoring"*

  A paper discussing the effect of recent technology improvements on the design of power quality monitoring systems. This article has been peer-reviewed and was presented at the 12th IEEE Mediterranean Electrotechnical Conference (MELECON) in Croatia in May 2004. Reference [111], reproduced on page 166.

A.2 Prepared and Submitted

The following four papers were produced from the material in this thesis:

- *"Effects of Sampling Rate and ADC Width on the Accuracy of Magnitude and Phase Measurements in Power Quality Monitoring"*

  This paper contains material also found in chapter 10. It examines the effects of the ADC quantisation noise on the recovery of harmonic magnitudes and phases.

  Submitted to the IEEE Transactions on Power Delivery, manuscript ID TPWRD-00173-2005. Reference [113], reproduced on page 170.

- *"System Requirements of Real Time Continuous Data Acquisition for Power Quality Monitoring"*

  This paper contains material also found in chapter 4 and chapter 5. Issues which need to be considered when establishing requirements for a power instrumentation system are discussed, in view of three broad configuration categories.

  Submitted to the Elsevier journal Electric Power Systems Research. Reference [109], reproduced on page 176.

- *"Implementation Considerations in Power Quality Instrumentation Design"*

  This paper contains material also found in chapter 11. The implications of advances in technology are discussed for different typical configurations and key requirements.

  Submitted to the IEEE Transactions on Power Delivery, manuscript ID TPWRD-00581-2005. Reference [112], reproduced on page 182.

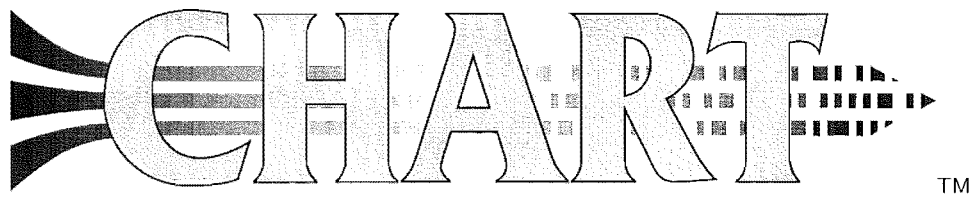
- *"Relevance of Microprocessor Advances for Power Quality Instrumentation"*

  This paper contains material also found in chapter 11. Modern microprocessor technology allows to concentrate instrumentation system functionality into fewer and general-purpose components, simplifying the design and reducing cost.

  Submitted to the IEEE Transactions on Power Delivery, manuscript ID TPWRD-00582-2005. Reference [108], reproduced on page 188.

A.3 Application Note

Synchronous Sampling with CHART [106].



Continuous Harmonic Analysis in Real Time

Application Note 001

Synchronous Sampling with CHART

Volker Kuhlmann

21 June 1996

Contents

1 Introduction 2
2 System Configuration 3
3 DSM Setup 3
4 GPS Receiver 5
5 DAPM Setup 5
6 Data Storage 6
7 Sampling Data and Displaying It 6
8 Tips 7

1 Introduction

This document gives a general overview for synchronous measurements using CHART. It assumes that readers are familiar with the structure and operation of CHART. For details about the internal operation of CHART, and how to operate the user interface of the CADU, please refer to [1,2,6,16]. The operation of the GPS receiver is explained in detail in its accompanying documentation.

The design of the CHART instrumentation system ensures a time synchronisation of all data acquisition channels of the same system by generating the sample clock centrally. This sample clock generated by the DSM is then shared by all sampling channels, with each DAPM providing three channels.

This system design provides a high level of synchronisation between the sampling channels, even if the time maintained by the central real-time clock is not very accurate. Whether a GPS receiver is connected or not, the precision of the synchronisation between channels remains the same. In this context please be aware of the difference between accuracy and precision. Precision is the closeness of the spreading of the actual values to the desired one, accuracy is the amount of spreading. If there is a lot of spreading, the accuracy is not very high; this says nothing about how close the actual (or measured) values are to the theoretical (or expected) one.

The design of the CHART system also provides means for synchronisation of sampling channels between different CHART systems. This is accomplished by a common and accurate time reference for each of the CHART systems. The GPS was chosen as the common time reference [1, 3–5, 16]. It is ideally suited for distributing accurate time information to geographically separated CHART systems. Its accuracy is in the order of a few hundred nanoseconds.

The GPS time reference is used to synchronise the beginning and ending of sampling between CHART units. It is also used to time stamp each sample (or packet of samples) taken. It is however not used to synchronise the taking of samples between CHART units, or to equalise the sampling frequencies between CHART units. In other words, the GPS time reference is used only for turning the sampling on and off, but not in between.

This application note explains the implementation of synchronous measurements using the CHART system. The example provided in this note has two CHART units, each with its own GPS receiver as time reference, as shown in figure 1.

The procedure described herein will draw heavily on features recently introduced into CHART and described in [6]. Please refer to this document for clarification of various terms and features referenced in this application note.

2 System Configuration

It is assumed in this application note that two CHART systems are used for measurements, and that they are situated geographically far away from each other[1]. Thus, a GPS receiver is required for each of these units.

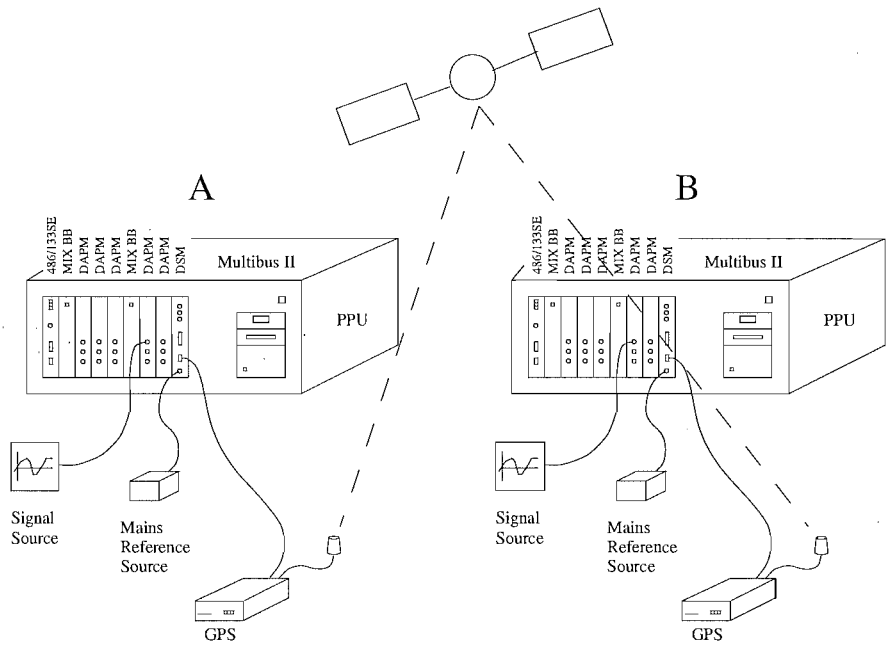


Figure 1: Two separate CHART units set up for synchronous measurements. Whether or not a mains frequency reference is connected is irrelevant for synchronisation, it has only an effect on the sampling frequency. For testing the synchronisation, both DAPM inputs should be connected to only one signal source.

When using CHART to carry out measurements, which include synchronous measurements, there are two types of units within CHART that need to be configured. They are the DSM and each DSP with the DAPM. Note: there are 3 DSPs on board each DAPM and in this note, the word DAPM is used to refer to each DSP within it. The DSM is responsible for generating the sampling pulses and the time information while the DAPM carries out numerical processing (such as FIR filtering, FFT) on the acquired data samples.

3 DSM Setup

General description of setting up the DSM for synchronous measurements.

[1] For testing the synchronisation between two CHART units, both units can be placed next to each other and supplied with the same known signal from a signal generator, as outlined in figure 1. Thus a known signal is digitised by a DAPM channel of each of the units. Except for the common signal source, there should be no other connections between the two units.

With most (if not all) measurements, the DSM is used as the centralised source of sampling pulses within each CHART unit. For synchronous measurements, this generation of sampling pulses has to be synchronised across multiple CHART units, i.e. between DSMs of different CHART units. This is achieved by setting up the DSM to make use of the GPS signal as the reference for starting or stopping the generation of sampling pulses.

The following lists individual DSM setup values, and what they have to be set to for synchronous measurements. All setup values are described in detail in [6].

The DSM must be configured as master DSM. The slave option is only used with multiple DSMs in the same CHART system. The bus drivers on the DSM driving the TSB (time stamping bus) need to be enabled by inserting the respective jumper. The default when shipping a DSM is to enable those bus drivers. For details refer to [7].

SRM mode. Either "Free running" or "Synchronised to mains frequency". This only affects the sampling frequency, not the time stamping of samples.

Sampling mode. This can be any of the available choices "Continuous sampling", "Burst sampling", or "Packet sampling". It only affects the number of samples taken, not the sampling frequency or time stamping.

Sampling Control. This must be "Deferred sampling". Only this allows a synchronised sampling start at a preset time. The beginning and end of the sampling can be set with "Start sampling time" and "Stop sampling time". The time and date is entered in local time. The local time for the CADU program can be set up with the TZ environment variable. For details refer to the CADU manual and the MS-DOS manual. For synchronous measurements, deferred sampling must be used and the starting time in each of the DSMs (in separate CHART units) must be set to the same time. Normally, the stop time is also set to the same time on all DSMs, as needed.

The FPGA configuration must be either "Continuous", or "ContBurstPacket". Sampling modes "burst" or "packet" require the latter FPGA configuration, "continuous" works with either.

With the current DAPM application "HarmAC", the number of vernier bytes must be 0. This allows a time stamping resolution of 1 s by using a 32-bit time value. The time stamping precision is still in the order of 1μs.

The GPS receiver type must be appropriately selected (currently, that means "MX4200").

After a DSM has been set up, it should be started. Sampling can not commence before the DAPMs and data storage are set up as well, see sections 5, 6.

Thus a DSM setup would typically be:

Master	→ yes
SRM mode	→ Synchronised to mains frequency
Sampling mode	→ Continuous sampling
Sampling Control	→ Deferred sampling
Start sampling time	→ Mon 16 Jun 2005 14:30
Stop sampling time	→ Mon 16 Jun 2005 17:30
FPGA configuration	→ ContBurstPacket
Vernier bytes	→ 0
GPS receiver type	→ MX4200

And the DSM status display window would typically show:

```
DSM on                YES
DSM master            YES
Sampling On           NO
Zcross detected       YES
SRM locked            YES
Mains frequency       50.02 Hz
1pps detected         YES
RTC locked            YES
GPS msg received      YES
Satellites visible    11
Satellites tracked    10
PARIO In              1110000
RTC time (Local)      Wed 13 Jun 1996 15:00:00
GPS time (GMT)        Wed 13 Jun 1996 03:00:00
```

4 GPS Receiver

A GPS receiver is connected to the DSM as time reference. The type of GPS receiver connected is selected in the DSM setup. This must be selected correctly if the DSM is to make use of the time information from the receiver.

A warm or cold start command can be issued from the DSM control window. Here the receiver can be initialised as well, although this should hardly be required because the DSM issues the initialisation commands automatically when necessary.

The following is valid for the MX 4200 receiver only. If there is a problem with tracking satellites, which can be seen on the status display, or by the LEDs on the front panel, it often helps to perform a receiver warm start. The receiver is not tracking any satellite if the yellow LED is solid. It is tracking at least one satellite if the green LED is flashing. If the green LED is solid, a sufficient number of satellites for accurate time recovery is tracked. If this does not solve the problem, a cold start should be performed. This clears all internal values and satellite information. A cold start is also advisable after transporting the receiver to a new location. Recovering from it should take about 15 min. Details can be found in the accompanying documentation [8–15].

5 DAPM Setup

In the DAPM setup, select the application "HarmAC", and set "Sampling Source" to "DSM". This instructs the application to pick up the sampling pulses and the time stamp from the DSM through the time stamping bus (TSB). For all synchronous measurements, regardless of the DAPM application, the sampling source must be set to DSM.

For any measurements, the DAPMs need to be loaded with the program required by the user, to carry out specific data processing. In addition to this, synchronous measurements with multiple DAPMs require that the DAPMs be at the same "position" when the sampling is initiated. This means that no DAPM must be running before the synchronous measurements commence. If they have been running, they need to be brought back to a common position by either reloading and reinitialising the program binaries, or restarting (i.e. stop then start) each DAPM.

A "restart all" command is available from the CADU which first issues stop commands to all processors (both DAPM and DSM), followed by start commands, effectively restarting all processors. Note that the above DAPM setup has to be done without the DSM generating any sampling pulses. This can be achieved either by having the DSM stopped so sampling is disabled, or preferably, by having the DSM wait for the start-sampling-time to arrive. Therefore, care should be taken that the "enable sampling" setup parameter is reset, or the start-sampling-time has not been reached, when using the "restart all" command.

Relevant items for synchronous measurements in the DAPM setups would typically be:

Application	→ HarmAC
Sampling source	→ DSM

After all DAPMs and DSMs have been set up, data storage can be configured.

6 Data Storage

Data storage is set up *after* all DAPMs are initialised, and *before* sampling is started by the DSM. Setting up data storage is described in [2]. In short:

Data storage is set up by clicking on the disk icon on the CADU. Fill in the fields. The location where the files is stored on the hub is displayed in the data storage setup window if it is reopened. It should be in /chart/project/username/data, where username is the name of the currently active user of the system, and data is the filename entered. Two files are created, one with the sample data, and the other with a description of the data file. Both files must always be kept together.

7 Sampling Data and Displaying It

The setup for the DSM, DAPM, and the data collection described in sections 3, 5, and 6 must be performed for each of the two CHART units separately.

By now everything is set up. It is advisable to check whether the deferred sampling time, or start sampling time, is still in the future...

When enough data has been collected, sampling can be stopped manually by using the respective DSM control, by changing the setup, or by making use of the stop sampling parameter which takes effect together with the start sampling time parameter when in deferred sampling mode.

Take care to close the data files properly by deleting the data storage entry with the CADU, otherwise the data will not be readable by the post-processing utility later.

The sampled data can be transferred from the CHART unit to a PC running postcess, using ftp. Postcess is the sample data display application.

When testing two CHART units for synchronous measurements, displaying both channels in the same window should show overlapping signal lines.

8 Tips

This section provides tips when using CHART for synchronous measurements. The following steps simplify the setup of the system and help to avoid mistakes when undertaking such measurements.

1. It is best to set up each system in the order: DSM, DAPM, data storage.
2. First of all, load the DSM with the appropriate software and enable the deferred sampling mode. The start time should be set far enough ahead to ensure that there is sufficient time to set up the DAPMs and the data storage. If there is insufficient lead time and the sampling starts before everything is set up, the whole setup process must be repeated from the beginning.
3. Activate and start the DSMs, which will wait until the start sampling time before generating samples. Check the operation of both DSMs with the CADU status display.
4. Load each DAPM with the appropriate application. If they have been loaded, "restart" each one of them. Be sure that the "sampling source" parameter is set to "DSM" before they are restarted.
5. Set up the data storage for the measurements.
6. When the start time is reached, the DSM will automatically start generating sample pulses on the TSB. The DAPMs relay these pulses to the RDCMs and process the acquired samples. The acquired and processed data will then be stored automatically, according to the data storage setup.
7. When the stop time is reached, or sampling is turned off manually from the CADU by stopping the DSMs, sampling and data processing cease.
8. Each of the data storage files needs to be closed before it can be accessed for post-processing purposes. To close the data storage file, "delete" each of the data storages in the data storage setup dialog box.
9. After the data storage has been deleted, the data file (*.dat) and the definition file (*.def) can be transferred for post-processing analysis and reporting, by using ftp.
10. If further synchronous measurements are desired, the start and stop times in the DSM setups need to be adjusted, all DAPMs restarted, and data storage set up again.

References

[1] John Axford. *CHART Technical Reference.* University of Canterbury, Christchurch, New Zealand, October 1994. Version 1.00.

[2] John Axford. *CHART User Manual.* University of Canterbury, Christchurch, New Zealand, October 1994. Version 1.00.

[3] Global positioning system information. United States Naval Observatory Automated Data Service. File gpssy, 941103.

[4] GPS time transfer information. United States Naval Observatory Automated Data Service. File gpstt, 941103.

[5] Volker Kuhlmann. A precision time base for synchronised distributed data acquisition. Diplomarbeit, not submitted, March 1995.

[6] Volker Kuhlmann. Sample pulse generation and time stamping by the DSM, and other improvements made in 1Q 1996. Technical report, University of Canterbury, Department of Electrical and Electronic Engineering, Christchurch, New Zealand, June 1996. Version 1.4.

[7] Volker Kuhlmann. Sample pulse generation and time stamping by the DSM, and other improvements made in 1Q 1996. Technical report, University of Canterbury, Department of Electrical and Electronic Engineering, Christchurch, New Zealand, May 1996. Version 1.3.

[8] Magnavox Electronic Systems Company, 2829 Maricopa Street, Torrance, California 90503, USA. *MX4200 Installation and Service Manual*, October 1991. R-7122B, No. 60232.

[9] Magnavox Electronic Systems Company, 2829 Maricopa Street, Torrance, California 90503, USA. *MX4200 GPS Receiver Programmer's Guide*, February 1992. R-7145A, No. 60222.

[10] Magnavox field service bulletin 4200-016: MX 4200 software release version B035 & version D035. 2829 Maricopa Street, Torrance, CA 90503, USA, April 1993. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[11] Magnavox field service bulletin MX4200-001: MX 4200 solid yellow LED. 2829 Maricopa Street, Torrance, CA 90503, USA, December 1990. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[12] Magnavox field service bulletin MX4200-002: MX 4200 reset commands. 2829 Maricopa Street, Torrance, CA 90503, USA, December 1990. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[13] Magnavox field service bulletin MX4200-006: MX 4200 common questions and answers. 2829 Maricopa Street, Torrance, CA 90503, USA, January 1991. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[14] Magnavox field service bulletin MX4200-015: MX 4200 software version B031 and version D031. 2829 Maricopa Street, Torrance, CA 90503, USA, April 1992. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[15] Magnavox product support bulletin MX4200-37: MX 4200 software version B35. 23860 Hawthorne Blvd., Torrance, CA 90505, USA, August 1994. (310) 791-2141; FAX (310) 791-6108.

[16] Allan John Vaughan Miller, Dennis Chuah, Michael Bernard Dewe, and Volker Kuhlmann. *CHART System Design Overview.* University of Canterbury, Christchurch, New Zealand, 1.00b edition, November 1994.

A.4 AUPEC 2001 Paper

Australasian Universities Power Engineering Conference (AUPEC), 23-26 September 2001, Perth, Australia [110].

SOME ASPECTS OF PRECISE SYNCHRONISATION OF DATA ACQUISITION FOR POWER SYSTEMS HARMONIC ANALYSIS

V. Kuhlmann M. B. Dewe C. P. Arnold

Department of Electrical and Electronic Engineering
University of Canterbury
v.kuhlmann@elec.canterbury.ac.nz

Abstract

To minimise losses in a power distribution network it is necessary to examine the power flow and perform a harmonic analysis. Design aspects of a multi-channel data acquisition system, which can operate continually, are discussed. Special emphasis is put on the accurate time-stamping of data when multiple acquisition units are operated in different geographical locations. The data acquisition system developed at the University of Canterbury, CHART, is given as an example.

1. INTRODUCTION

The tools available for power system harmonic analysis are very limited. This makes it difficult, if not impossible, to obtain useful results. Therefore, the researchers at the University of Canterbury's Department of Electrical and Electronic Engineering decided it was necessary to build their own equipment [1], in order to be able to capture sufficient data to understand the phenomena. This has proved to be a major undertaking.

In order to analyse harmonics on power systems, it is necessary to monitor many variables at different geographical locations simultaneously. This puts severe constraints on the type of monitoring and analysis equipment which can be used.

The amount of data which must be recorded at the same time can vary considerably. At each geographical location many signals (current and voltage usually) may need to be acquired and processed simultaneously. It is important that the processor unit should have flexibility in the number of data channels it can handle and that it be able to be extended, within reason, according to needs.

It is not sufficient to just obtain "snap-shots" when analysing power system harmonics. The data must be acquired continuously in real-time and precisely time-stamped. Further, to reduce the amount of data which needs to be stored, the data should be brought together from different channels and real-time processing performed using some hierarchical layers of processing. Accurate time-stamping is required so that the data from units at different locations can be combined for further post-event processing. The GPS system provides an ideal means of obtaining accurate time-stamping for all the acquired data.

A difficulty with using temporary equipment in a switchyard is possible problems of isolation. This can be overcome to a large extent by using fibre optics to transmit the data from the ADC units in the yard to the processor unit in the control room. The ADC units also provide a means of transmitting sampled data rather than signals to the processor.

The dynamic range of the monitoring equipment is also an important consideration. While voltage levels are reasonably consistent, current can vary from zero to very large values in a short period of time. To ensure optimum signal resolution in the A/D conversion process, a dynamic variable gain front end needs to be introduced. A 12 bit dynamic range would allow scaling by 4096 different values.

A major use of the equipment is to aid research into harmonic problems. Therefore the equipment must be flexible. The software should be able to be modified by downloading new algorithms into the system, and the display of results should be able to be customised. This allows the expensive hardware to be used for different applications. Further, the storage of data should be flexible to allow for resolution and accuracy as well as the nature of the data formats. As indicated above, using real-time data processing and analysis reduces the amount of data which needs to be stored for future analysis. A final requirement of equipment of this nature is that the hardware can be upgraded incrementally.

This paper describes some of the problems and solutions of power system harmonic data acquisition.

2. SAMPLING SYNCHRONISATION

For multi-node network analysis it is necessary to sample synchronously at all locations involved. Synchronisation to an absolute time reference is not critical, it is more important for the distributed instruments to be synchronised to each other. To achieve this, a master clock could be set up within one of the instruments involved, and its time could be transferred to the other instruments. However, this kind of time transfer is not trivial when the instruments are located far from each other. The propagation delay over any kind of link would by far exceed the desired accuracy of the synchronisation, not to mention the problems associated with establishing any link at all.

An alternative is to use the Global Positioning System, GPS, as time source for each of the instruments involved [5]. This would synchronise instruments to absolute time, which, although not essential, might prove beneficial for a particular application. Time transfers between instruments are no longer needed.

According to Shannon's sampling theorem, a minimum of 2 samples per cycle of the highest harmonic of interest are necessary to accurately describe a waveform. For the 50th harmonic of a 50 Hz fundamental this is a minimum of 2 samples every 400 μs, or a sampling frequency of 5 kHz. To establish power flow in an electrical network, relative phase angles are important. By having time-stamping accurate to 1 μs, this gives a phase error of $<1^\circ$ at the 50th harmonic. An accuracy of 1 μs is within the means of the GPS.

Although synchronisation to absolute time is not a primary requirement, it is easier and cheaper to implement than time transfers between instruments.

This effectively means that one master clock would be provided in each measurement location, with all master clocks being synchronous to within the accuracy of the GPS. The master clock should control each of the ADCs in the system, ensuring that there is no time difference in the sampling between the ADC channels.

The measurement instrument would most likely be deployed in switch yards, which have a high level of electrical noise as well as a danger of high step voltages. As the signals to be measured can be some distance away from the measurement instrument, the ADC unit could be integrated into the main unit and the signal transported from the source to the main unit, or the ADC unit could be located close to the signal source. The latter choice is preferable because it avoids carrying analog signals through a noisy environment, because an already digitised signal is carried instead. A drawback is that the ADC clock signal needs to be carried from the main unit to the ADC unit, thus requiring a full-duplex link, but this is not a significant disadvantage compared with the benefit of avoiding signal distortion.

The link between the main and ADC units can be implemented with coaxial or fibre-optic cable. The shielding of coaxial cable may still be insufficent in a switch yard environment. Fibre-optics would be immune to switch yard noise, and also provide electrical isolation between units. This also weighs in favour of implementing ADCs remotely, because it provides reliable isolation between the ADC units and the main unit.

For calculation of harmonic levels FFT windowing issues need to be taken into account. These occur when the number of samples taken per cycle of the fundamental differs from the predefined number. Compensating for this during the transformation is computationally expensive and takes DSP resources away from the primary task.

Another approach is to vary the sampling frequency according to actual mains frequency such that there is always the same number of samples per fundamental [5]. This should keep the errors introduced due to windowing at a suitably low level, without taking up CPU time. The mains frequency would need to be tracked continuously and the sampling frequency adjusted accordingly.

3. HARDWARE REQUIREMENTS

Apart from the hardware required for sampling synchronisation, which is listed above, and the hardware for time-stamping introduced below, processing power must be sufficient for the task. Some of the other resources usually found in a computer system are also required.

The processing power in the system is influenced by the chips available at the time and their cost, the practicality of combining a number of chips into the same system, the processing requirements of the intended application, and any spare CPU resources desired for possible future applications. The minimum would be to use a multiplexed ADC stage and one processor for analysis of all input channels.

For time-stamping with an accuracy and resolution in the order of 1 μs, a pure software solution is not adequate. DSPs like the TMS320C3x from Texas Instruments can execute approx. one instruction in that time, which is not nearly enough to process an interrupt in an interrupt service routine. Latencies in processing of time information may not be equal between all sampling channels, and the response time typically guaranteed by a real-time operating system is too long for

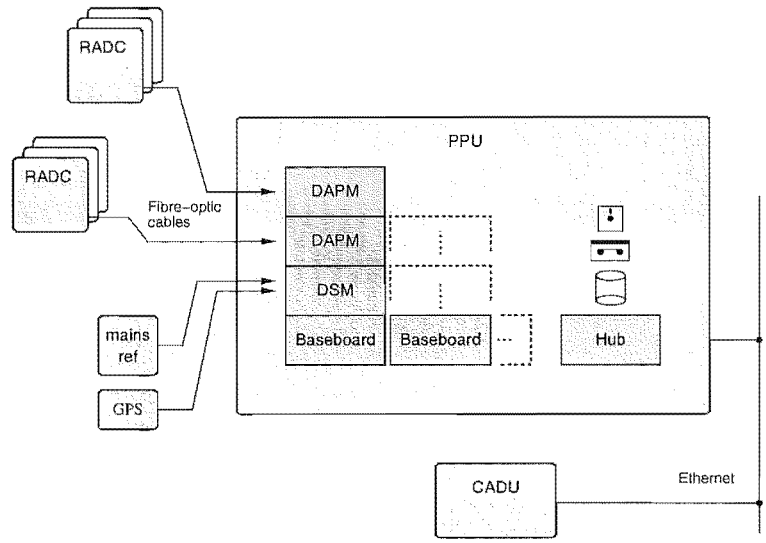


Figure 1: Overview of one CHART system with remote ADCs, mains frequency reference, GPS receiver, main unit/hub, and control and display unit (CADU).

time-stamping at this accuracy.

When software is not fast enough, implementation in hardware is in order. This can take a considerable load off the CPU, and guarantees that there are only minimal time errors between channels. Tying the time-stamping in with the sampling further improves coherency between the two.

The following hardware would have to be implemented:

- DSP with program and data memory,
- sample clock and data I/O between DSP and ADC,
- sample clock, time-stamping and program/data I/O between DSP and main system (hub).

The program memory for the DSP shouldn't be fixed so that user-defined application programs can be executed. Furthermore, I/O with the GPS receiver, sample clock generation, and tracking of actual mains frequency needs to be implemented; this should be on a separate card as it is only needed once per main unit. To reduce the amount of hardware, one DSP could service more than one ADC, or more than one DSP could be combined per card, sharing the I/O circuitry to the main unit.

Ideally, a platform for development of this custom hardware would fulfil the following requirements:

1. A hardware platform supported by multiple vendors, and for which specifications are controlled sensibly. This hardware platform must:
   (a) have a sufficient rate of throughput
   (b) have multi-CPU capabilities
   (c) offer a minimum card size for user hardware
   (d) offer a sufficient number of slots for cards
   (e) allow user-defined signals to be carried between user-designed cards
2. An operating system which is available for the above hardware. It should have a certain customer acceptance, and it must:
   (a) be multi-tasking
   (b) offer real-time features typically required for process control
   (c) incorporate network capabilities currently in common use

The choice of hardware should be directed towards commonly available digital systems, because the investment in hardware designs is huge and not easily portable. For harmonic analysis on a desirable number of channels simultaneously, desktop PC hardware does not support sufficient bandwidth, card size or card numbers. The alternative are industrial 19 in rack systems, although they incur a noticable increase in cost.

The rapid change in underlying technology, e.g. processing speed, memory cost, microchip integration level, or power consumption, can quickly change the constraints of an instrumentation system.

With advances in microchip technology, a solution which is likely to be more cost-effective is to merge the DSP stage with the ADC stage and having both separated from the main unit. The interface between DSP and main unit would have to be designed to cater for the

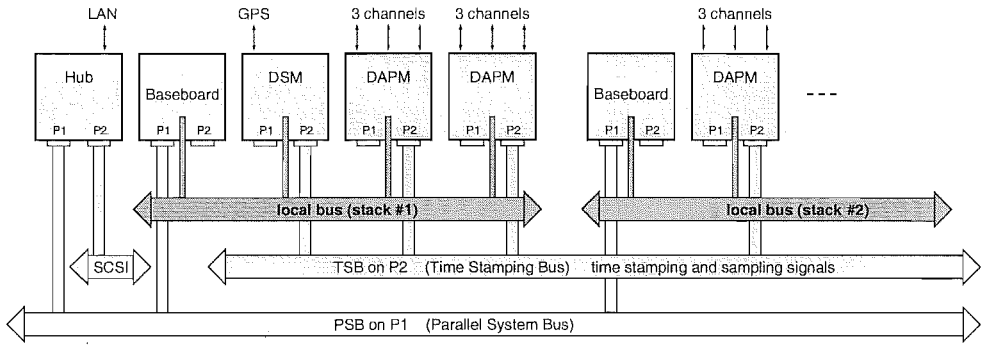


Figure 2: Data flow between the cards of the CHART system.

transfer of time-stamping signals, DSP application software, and DSP control from the main to the remote unit, and for the transfer of status information and packets of time-stamped data back to the main unit. The largest advantage of this design would be the much easier constraints on card size and slot numbers.

4. REAL-TIME OPERATING SYSTEM

The choice of operating system is not clear for large volume data capture and processing systems.

General purpose operating systems have the advantage of being optimised to have a very high throughput, but the time taken to perform specific tasks is not guaranteed. On the other hand, real-time operating systems guarantee the time for an operation but throughput is generally sacrificed.

Real-time operating systems are developed for specific applications and are not so common as general purpose operating systems. This makes them more prone to implementation errors and they are not so well supported. A further problem is that, should operating time or memory limits be exceeded, the system collapses, unlike a general purpose operating system which may lose some data but will keep on going.

The CHART system, described in more detail in the next section, uses a real-time operating system. While this was the correct choice when the project started, it is now debatable whether this is necessary given the much higher processor speeds now available.

5. THE CHART SYSTEM

The CHART system has been designed to find the harmonic levels of as many as 32 power system voltage and current signals on a cycle-by-cycle basis continually in real-time. Hence the acronym CHART — Continuous Harmonic Analysis in Real-Time. Harmonics are computed for every fundamental cycle continuously and in real-time, and can be displayed as they occur. CHART can also compute and display harmonic impedance and power of a monitored bus. Figure 1 gives an overview of one CHART system. Generally, two or more systems are used in conjunction.

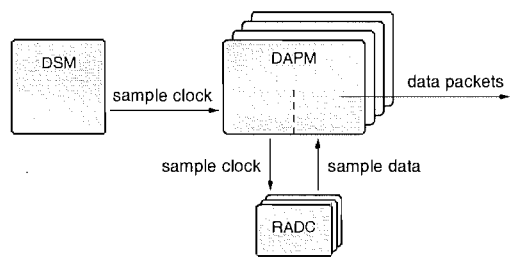


Figure 3: Generation and propagation of the sample clock: from the digital services module (DSM) to the remote ADC (RADC). Sample data coming back from the remote ADC is time-stamped and put into packets by the data acquisition and processing module (DAPM).

CHART samples signals at a rate which allows the 50th harmonic to be recovered. This quantity of data is not onerous and can easily be handled by independent ADCs at the point of monitoring and then transmitted to a central location by a fibre-optic link. However, harmonic levels are computed using the FFT to find the discrete Fourier transform (DFT), which can be efficiently implemented using DSPs. For the number of channels required it was necessary to develop a two channel Multibus II based DSP board. A typical system consists of four or more boards which can be easily increased to improve processing capabilities and increase the number of channels. The bus configuration for handling the data transfer between the data acquisition and processing modules (DAPM), the hub and

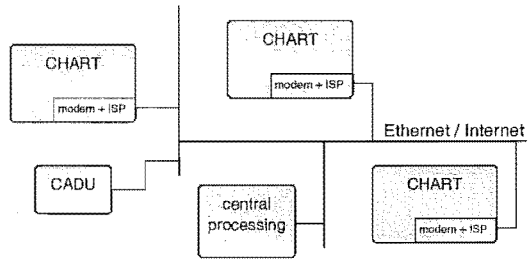


Figure 4: Using multiple CHART systems for synchronised distributed measurements.

the baseboard are shown in figure 2. In figure 3, the means by which the generation of the sample data is controlled and propagated to the DAPMs is shown. An additional advantage of using DSPs is that many different harmonic analysis algorithms can be tested, because the system is designed to allow simple loading of executables into a DSP's program memory.

CHART incorporates a GPS satellite time reference. By using several CHARTs at geographically separate locations on the network, the system has the inherent capability of enabling precise simultaneous measurements of power system parameters. Where data is being sampled at different geographic locations, the CHART units must be interconnected by a network as shown in figure 4. Only one control/display unit is needed for all mesurements systems. For analysing data from two or more CHART units, a central processor is required somewhere on the network.

The front-end processing of the sampled data happens on the DSPs. In the case of harmonic analysis, harmonic information is extracted from the time domain data, arranged into packets and time-stamped. A further stage of processing by the main unit can be used to implement filtering and detection according to user-defined criteria, e.g. harmonic levels exceeding a preset limit. The resulting data is then stored for record keeping and further analysis, and can be displayed to the user. A detailed description of the structure of the software for the DSPs and the hub can be found in [2, 3].

Without this two-stage processing, many gigabytes of data can easily be collected. With it, the amount of data collected does not exceed the capacity of the system. This has the added advantage that the retained data is focused on the signals of interest, without the need for elaborate further processing.

6. CONCLUSIONS

For continuous precise time-stamping with high resolution, it is necessary to carefully consider issues of data flow and its matching with time information. A purely software-based solution is not sufficient for high-end requirements, although advances in CPU technology will reduce the need for specialised time-stamping hardware in the future.

7. ACKNOWLEDGEMENT

The authors wish to acknowledge the financial contribution of Transpower NZ Ltd towards the development of the CHART system.

REFERENCES

[1] S Chen and M B Dewe. System and hardware considerations of a flexible multi-channel real time data acquisition system for power quality assessment. In International Conference on Harmonics and Quality of Power [4]. Las Vegas, Nevada, USA.

[2] S Chen and M B Dewe. Virtual operating system design in a flexible multi-channel real time data acquisition system for power quality assessment. In International Conference on Harmonics and Quality of Power [4]. Las Vegas, Nevada, USA.

[3] Shiun Chen. *Analysis of HVDC Convertor Transformer Core Saturation Instability, and Design of a Data Acquisition System for its Assessment.* PhD thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, January 1996.

[4] *The 7th International Conference on Harmonics and Quality of Power — ICHQP*, October 1996. Las Vegas, Nevada, USA.

[5] Allan J V Miller and M B Dewe. Multichannel continuous harmonic analysis in real-time. *IEEE Transactions on Power Delivery*, 7(4):1813–1819, October 1992.

A.5 MELECON 2004 Paper

Mediterranean Electrotechnical Conference (MELECON), Dubrovnik, Croatia, 12–15 May 2004 [111].

Impact of Processor Evolution on Synchronous Measurements

V. Kuhlmann, M. B. Dewe, C. P. Arnold
University of Canterbury, Christchurch, New Zealand
v.kuhlmann@elec.canterbury.ac.nz

Abstract— **In the past, data acquisition systems for power quality monitoring required expensive custom-designed hardware and software. With recent advances in standard PC components, this is no longer the case. This paper discusses requirements for modern instrumentation systems, and the effectiveness of standard CPUs and operating systems in fulfilling these requirements. A CPU register and interrupts can be used for sample clock generation and time stamping. Commercially available single-board computers are suitable for ADC front-ends. Modern CPUs include signal processing instructions previously only found in DSPs, and general-purpose operating systems such as Linux provide all the necessary software features.**

I. Introduction

Advances in processor performance bring with them new opportunities for instrumentation systems in terms of size, cost, and bandwidth. Custom-built hardware can be replaced with software and standard components. This paper examines how these advances affect instrumentation systems for power quality monitoring in particular.

The main requirements for power quality monitoring are continuous measurement and analysis of voltage and current vectors in power distribution systems. Of particular interest are harmonic components. A sampling rate of 5 kHz is sufficient for the 50th harmonic magnitude (50 Hz system), which does not place a high demand on hardware. The recovery of phase angles for harmonic power flow calculations with an accuracy of $1° - 10°$ is more demanding, especially for the higher-order harmonics. An even higher demand on equipment is placed when transients are to be examined, but this is not considered in this paper.

It is essential that the sampling of input values occurs simultaneously on all inputs, so the resulting values, which may for example represent several 3-phase systems in a switch yard or across a power network, can be correlated with each other. The upper limit of allowable time error between inputs is determined by the application, and is typically in the order of 1 µs.

Measurements at geographically different locations also require the same accuracy limits, to allow, for example, harmonic power flow calculations. Instruments need to be closely synchronised over large distances. If the instrument is to be used for monitoring, it needs to have sufficient bandwidth in all processing stages to handle a continuous flow of data. A decade ago, this required a lot of dedicated hardware for synchronisation and many DSPs to maintain the throughput. Substantially reduced hardware requirements translate into cost savings and make permanent deployment economical.

Isolation requirements, like those in a high-voltage switchyard environment, can also put further constraints on instrumentation system design.

A data acquisition system can be broadly described as having the stages

$$sensor \rightarrow ADC \rightarrow processing \rightarrow storage \rightarrow analysis$$

plus a suitable control mechanism for the sampling, the complexity of which heavily depends on bandwidth and accuracy requirements. The requirements of the ADC stage with respect to power quality monitoring are discussed in [1]. This paper discusses some of the key processing issues.

II. Processor Architectures

The trend to more complex processors and higher clock speeds is continuing, with both resulting in increased processor performance. Special instructions for signal processing, which were found only in DSPs 10 years ago, are now included in desktop processors like Athlon or P4. There is no longer a stringent requirement for using DSPs for power quality computations.

Architectural features like caching, pipelining, and execution concurrency were introduced to improve overall performance. These improvements however had a negative effect on interrupt latency [2]. After continuous speed improvements, latency is in the same order as it was 20 years ago. Execution times of code parts have become non-deterministic and are largely statistical. If the processor is used for time-critical tasks, any non-deterministic execution of CPU instructions becomes irrelevant if the amount of indeterminism is less than the timing accuracy required. The following section shows that this is now the case with modern processors.

III. Sample Timing Issues

The first step for any data acquisition system is to convert an input value to a digital representation. This sampling process is initiated by a signal to the ADC (analog-to-digital converter) stage. Depending on the application, a data acquisition system may have many input channels, each of which has an ADC. Synchronous measurements require that the sampling on each input occurs at precisely controlled instants. The maximum allowable uncertainty for this depends on the particular

measurement being undertaken, and is often much shorter than the sampling period.

For power quality monitoring, the time stamping accuracy of the sampling typically needs to be in the order of 0.1–1 µs. Harmonic power flow calculations compare phase angles at different nodes of the distribution network. 1° at the 50th harmonic is approximately equivalent to 1 µs. The samples need to be time-stamped with an accuracy better than 1 µs.

For a 50 Hz system, a sampling rate of 5 kHz would be sufficient to capture the 50th harmonic, but it is advantageous to use a power of two of the fundamental, e.g. 6.4 kHz (128 points at 50 Hz, no oversampling).

Synchronisation of sampling is achieved by clocking all ADCs with the same signal. The ADCs must have a known time after which the result appears at the outputs. The following tasks need to be performed:

1) sample clock generation
2) sampling at a multiple of the fundamental
3) time stamping

Sampling at a multiple of the fundamental frequency is not strictly necessary, but does significantly reduce FFT computational requirements by eliminating windowing issues.

It should be possible to perform all these 3 tasks by utilising a 32 bit counter register provided by the CPU. The counter is free-running with a suitably high frequency, about 50–100 MHz (or the clock frequency of the CPU), and pre-loaded with a value which, when decremented to zero, denotes an interval between samples. When the counter reaches zero, a sampling signal for the ADCs is generated, the counter is reloaded, and the ADCs are read.

The sampling frequency is adjusted by changing the initial counter value. The actual mains frequency can be measured by feeding a mains-derived and low-pass filtered zero-crossing signal into an interrupt input of the CPU. The mains frequency can be derived from the initial counter values and the number of times the counter has reached zero. Alternatively, the mains frequency can be assumed to be fixed. Calculating the mains frequency is not time-critical and should be performed outside the interrupt service routine. Using hardware counters for generating a mains-synchronised sampling clock is described in [3].

Time-stamping is performed by generating another interrupt by the one-second output of the time receiver. In the interrupt service routine, the above counter needs to be read and the value stored. This allows the determination of the number of counter decrements per second, which can be used when reading the ADCs to work out the number of counts which have elapsed since the beginning of the second.

Of critical concern is the timing and latency of the interrupts, but a number of factors can be used to advantage. All counter calculations only involve simple integer calculations and need not be performed inside the interrupt service routines. The order of operations inside the interrupt service routines should be considered carefully.

The interval of the 1 pps signal can be considered constant. The counter frequency will drift, but not change abruptly. This interrupt should be assigned highest priority. If the counter frequency ($\sim$ CPU clock frequency) is calculated to have jumped by an unreasonable amount, the result should be discarded for this second.

Likewise, the zero-crossing interrupt may be masked by a higher-priority interrupt for a short time, <20 µs. The software can cope with this, combined with the assumption that mains periods do not change significantly for single cycles. A simple suitable low-pass or averaging filter should be used as well, which also eliminates jitter generated by noise.

The sample clock interrupt is non-critical. When reaching zero, the counter generates an external clock signal and reloads, without software. The ADCs can be read any time before the next sample clock pulse.

An 80 MHz Power PC CPU has an interrupt latency in the order of 1–2 µs. 400 MHz versions are available. A large part of the latency is deterministic and can therefore be compensated for. An accuracy of <1 µs can be achieved without great difficulty.

As a test of the time required for time-consuming calculations, benchmarks with the commonly used fftw library [4] have shown that even a relatively slow 450 MHz Pentium 3 CPU can calculate a 128 point complex FFT in under 15 µs. For 6 channels, this would be <1/200 of a mains cycle. Filter or THD computations are of the same order of complexity as those of FFTs.

A standard PCI bus is clocked with 33 MHz. A 32 bit word can be transferred in one bus cycle in burst mode, which has an overhead of a few cycles. A single bus access takes approximately 4 bus cycles. This throughput is sufficient for at least 6 ADC channels.

It is therefore possible to perform sample clock generation and time stamping in software with the help of a CPU register, while only placing a small to moderate load on the CPU, leaving room for processing of the data.

IV. System Architecture

Power quality monitoring places a number of additional constraints on instrumentation systems. Switch yards, for example, are an electrically noisy environment and can produce potentially large voltage differentials. Signal sources are likely to be located many metres from a control room. This makes it impractical to carry signals to ADCs. Instead, ADCs must be located close to the signal sources. If signal data is carried digitally to the control room, it is relatively immune to noise and further signal degradation is prevented.

For measuring voltages and currents of a 3-phase power distribution system, 6 ADC channels are needed. At a sample rate of 50 kHz (8* oversampling) and an ADC width up to 16 bit, the resulting 600 kbyte/s data rate is well within the capabilities of current CPUs, with computing power to spare.

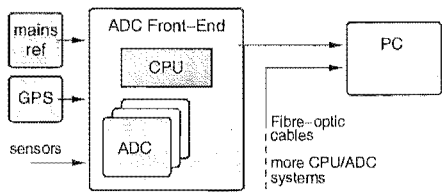


Fig. 1. A typical configuration of the proposed sampling hardware.

Bandwidth becomes tighter if oversampling and an anti-alias filter is used. The advantages of doing this are a reduction of computational requirements further on, and a reduction in analog front-end filter component size.

The more processing can be performed in the front-end, the less bandwidth and computing is needed further on at the central processing and analysis stage, however the particular application will set a limit on how much can be left to the front-end.

Tasks which also need to be performed include auto-range control of the inputs, and time stamping of the sample data (see previous section).

Depending on circumstances, a simple noise-reduction filter can be implemented. Compensation for non-linearities of the sensor makes post-processing of data convenient. This can be sped up by using a simple pre-computed look-up table. A 16 bit ADC and 32 bit floats translate into a 256 kbyte table size, which is small by today's standards. It is not essential to compensate at the ADC front-end, but doing so presents filtered and corrected data as output of the ADC front-end, which reduces complexity and processing at later stages.

At the minimum, it is desirable that the ADC front-end provides at least slightly noise-filtered data which is time-stamped with sufficient accuracy and corrected for sensor non-linearities.

A. *Front-end design*

Instead of making custom hardware, this front-end CPU system can be based on one of the commercially available single-board computers. Besides a CPU and memory, these boards often provide at least one PCI bus and various peripheral interfaces, including RS-232C, network, or IDE hard disk. Intended for use in embedded systems, the form factor is comparatively small, and a low-power version of the CPU is used. CPU power is therefore likely to be a little lower than in top-end desktop PCs. In other words, "a laptop with PCI slots and no case" is a reasonable description of the desired features.

Using, in essence, a PC for this ADC front-end provides further benefits. Single-board computers are commercially available at comparatively low cost. Most of the software development can be done on a PC. Software development systems are much more advanced for PCs and standard tools can be utilised. Development for embedded systems is intrinsically more difficult and requires expensive specialised tools.

Some custom hardware will need to be interfaced to this CPU, which is easiest when implemented as a PCI bus

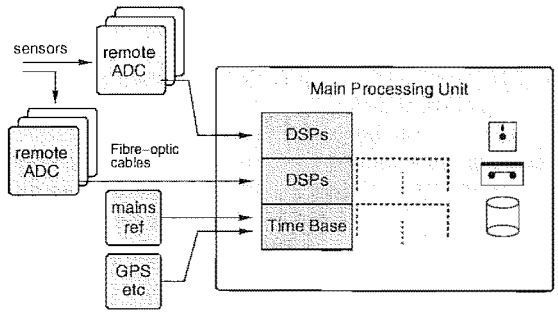


Fig. 2. A typical configuration of the previously developed CHART instrumentation system, which has a hardware time base.

card. The custom hardware comprises the analog input stages, scale adjustment for auto-ranging, and ADCs, once per input channel, as well as those parts of sample clock generation and time stamping circuitry which can not yet be implemented in software. It is desirable that all required custom hardware is integrated on one or more PCI cards. If more than one card is required, all cards should be identical in design. Depending on the particular components used, it is possible to fit everything into a standard small form-factor PC enclosure.

For measurements involving more than one geographical location, accepted practise is to use a commercially available satellite time signal receiver as time base, e.g. GPS, GLONASS, or Galileo. These devices supply coarse timing information (1 s and up) via an RS-232C interface, and a digital one-pulse-per-second output, which must be interfaced to the time-stamping circuitry. If the time-stamping is implemented in hardware, the 1 pps signal will have to be supplied to all PCI cards via interconnecting cables.

Flash memory is now available in capacities which are high enough to store the operating system for the front-end CPU, all the application software, and correction tables for a large number of sensors. The system needs to boot from this flash memory. Some single-board systems provide an IDE hard disk interface that, when combined with an IBM Microdrive (flash memory which behaves identically to a hard disk), becomes a storage solution any operating system today should be able to access easily.

B. *System configuration*

If more than one front-end ADC system is needed at the same location, the same satellite time source can be used to provide sychronisation. This becomes more economical with a larger number of channels per front-end. It is not as straightforward as having only one time source in the base station. However, the need for very time-critical transfer of ADC clock signals from the base station to the front-end disappears.

Data needs to be transferred from the front-end to the base station with some kind of network. The internet provides an array of hardware and protocols, the obvious choice being ethernet. A 100M ethernet link provides about 10 Mbyte/s throughput, which is more than suffi-

cient for power quality applications. PCI cards with fibre-optic 100M ethernet interface are readily available, and any operating system should be able to handle internet protocols.

An overview of a typical deployment of the proposed system is shown in figure 1. For comparison, figure 2 shows a similar system previously developed at the University of Canterbury, based on a hardware time base.

An alternative, simplified system design can be built from a desktop PC, acting as both front-end processor and base station for control, storage and analysis. This is sufficient for a limited number of channels, with more channels as processors become faster. Implementing all custom hardware on a PCI card allows its dual use. This is also an excellent way to develop the software for the custom hardware. The issues for interfacing time signal receivers are identical to those in separate front-ends. Multiple desktop systems can be used if necessary. This constellation can be seen as the budget version of a top-end instrumentation system.

Power supply issues need to be resolved. Batteries can be used for continuous operation up to a certain limit. For permanent operation, a suitable isolated power supply must be made available.

V. Software Considerations

The choice of software for the base computer is reasonably straightforward. None of the tasks are time-critical, and communications are via the internet, which in itself provides a certain amount of buffering. Desktop CPUs provide ample power for number crunching, with AMD being slightly more efficient for floating point operations than Intel. Multi-CPU systems are available. A general-purpose operating system is well suited. Linux has proven to be reliable and would be a good choice for a dedicated instrumentation system. In particular, it enables users to log in remotely, which allows full use and control of the instrumentation system without having to be on site.

For the ADC front-end, a real-time OS like vxworks could be used. The drawback is that although these systems guarantee a response within a certain time, their average throughput can be quite low. When resources (other than CPU time) are depleted, they simply stop working. In contrast, general purpose OSes continue, but at a much lower speed. This means that they can no longer perform the intended task adequately, but they still allow the user to log in and fix the problem. This is a strong argument for an autonomous system which can be operated remotely.

For a data acquisition system, the real-time requirements are rather simple. With an external signal, counter or ADC output values need to be saved within a short time frame. Their processing is not time-critical. Saving values can be performed easily within an interrupt service routine, and modern CPUs provide adequately low interrupt latency. It is therefore not impossible to run a general-purpose OS.

A general-purpose OS can be run on a real-time kernel, e.g. [5], but this is much more sophisticated than is required. Real-time extensions for Linux exist as well, and are commonly offered commercially with hardware. A sufficient solution is to modify Linux to add a fast interrupt routine. The availability of the source code makes this possible, and versions for embedded systems already exist.

Depending on desired flexibility of the front-end, different programs can be run, possibly including a user-supplied component. The user-component can be implemented via a shared library API. On a general-purpose OS, shared libraries are commonly used, and can be loaded by the main program any time after startup.

Software for storing the programs, and exchanging control information, programs and data with the base system, are already taken care of by a general-purpose OS, simplifying that part of the design considerably.

If this instrumentation system is to be connected to the internet, security considerations need to be taken very seriously. This becomes paramount if the front-ends are connected to the base PC via the internet. The encryption which will need to be used for this incurs a non-negligible CPU load penalty, which must be considered. Experience shows repeatedly that internet security issues have not received appropriate attention.

VI. Conclusion

A single modern CPU can now be used for continuous synchronous measurements of voltage and current vectors in power distribution systems, instead of expensive hardware-based solutions. Utilisation of standard PC componentry and a general-purpose operating system reduces design complexity substantially.

A key issue is the accuracy of time stamping of the samples. A 32 bit counter register in the CPU can be used to time-stamp data and control the sampling frequency.

Single-board computers are suitable for implementing cost-effective ADC front-ends. Ongoing improvements in processor performance will make more data processing possible in the front-end.

The level of performance of modern processors is sufficient for the time-critical tasks of sample clock generation and time stamping, while still handling the data stream. A number of different configurations are possible. Low-cost data acquisition systems can be built, allowing a greater use of informative quality monitoring of power systems.

References

[1] S. Chen, "A quantitative analysis of the data acquisition requirements for measuring power quality phenomena," *IEEE Trans. Power Delivery*, vol. 18, no. 4, pp. 1575–1577, Oct. 2003.

[2] H. C. Torng and M. Day, "Interrupt handling for out-of-order execution processors," *IEEE Trans. Comput.*, vol. 42, no. 1, pp. 122–127, 1993.

[3] V. Kuhlmann, "Fault location on HVdc lines," Master's report, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Feb. 1993.

[4] [Online]. Available: http://www.fftw.org/

[5] J. Odagiri, N. Yamamoto, and T. Katoh, "Porting EPICS to L4-Linux based system," in *Proceedings of the 2001 Particle Accelerator Conference (PAC 2001)*, vol. 2, 2001, pp. 1183–1185.

A.6 Sampling Rate and ADC Width Paper

Submitted to the IEEE Transactions on Power Delivery [113].

Effects of Sampling Rate and ADC Width on the Accuracy of Magnitude and Phase Measurements in Power Quality Monitoring

V. Kuhlmann, A. M. Sinton, M. B. Dewe, C. P. Arnold
University of Canterbury, Christchurch, New Zealand
volkerkuhlmann@gmx.de

Abstract— **This paper examines the effects of the ADC quantisation noise on the recovery of harmonic magnitudes and phases in order to determine the noise limits of instrumentation system design for power systems monitoring. We show by theory and simulation that magnitude and phase errors are independent of harmonic order, that magnitude error is independent of harmonic amplitude for sufficiently large signal to noise ratios, and that phase error is inversely proportional to harmonic amplitude. The noises may be reduced by increasing the ADC width or the transform length. A 12 bit ADC gives sufficiently accurate results for typical power quality applications.**

Index Terms— **Analog-digital conversion, Quantization, Power quality, Monitoring, Signal sampling, Discrete Fourier transforms, Digital filters, Digital measurements**

I. Introduction

DESIGNERS of instrumentation systems are interested in the extent to which each part of the system contributes to the final error. Harmonic analysers report the magnitude and phase for each harmonic order, therefore the error for these instruments is the differences in both magnitude and phase between the values reported by the instrument and the actual values being measured. A typical structure of a harmonic analyser is shown in figure 1. The final error is mainly made up of contributions from

- ADC (analog-to-digital converter) quantisation
- ADC non-linearities
- Analog electronics non-linearities

and is also influenced by

- Transform length (the length of the FFT, fast Fourier transform)
- Oversampling ratio
- Number of sample points (which is equal to transform length times oversampling ratio)
- Digital filter (downsampling filter) characteristics

This paper describes simulations of the errors inherent in the ADC quantisation with respect to their effect on harmonic analysis. The results provide designers of such instruments with information about the hardware required for a desired level of accuracy, and allow customers to assess the significance of results. Average and maximum errors are of interest for both magnitudes and phases.

The data acquisition stage (ADC) and fast Fourier transform (FFT) performance are simulated with particular emphasis on phase recovery. Phase information is important for establishing harmonic power flows. The simulation variables are sample rate (samples per fundamental period) or FFT length, ADC width (bits), the ratio of harmonic to fundamental amplitudes, and the harmonic order. Realistic known input values are fed into the simulation model, and the errors are calculated as the differences between the input and the output values. An ideal ADC is assumed for the simulations; other distortions resulting from non-linearities, sample and hold windows, etc are not considered.

Fig. 1. Components of a harmonic analyser. The digital low-pass filter is present if oversampling is used.

Measuring current harmonics on power distribution lines presents the problem of a largely varying signal amplitude. For line currents it ranges from near 0 (no load) to typical full load levels, whereas the voltage signal is practically constant. Fault levels are higher for both currents and voltages. Although in the case of a fault the harmonic content is not of primary interest, capturing the fault waveform is of interest for fault location applications. If transient capturing is not part of the objective of the instrument, the higher amplitudes of transients need not be considered. Automatic gain adjustment can be used to respond to changes in input signal amplitude, but has a number of disadvantages [1], [2].

Resolution issues in detecting the harmonics are a further potential problem, because the amplitude of the harmonics may be orders of magnitude lower than the amplitude of the fundamental. Filtering out the fundamental causes the relationship of harmonic amplitudes to fundamental amplitude to be lost, and requires sharp filters which are expensive to implement with analog components. The simulations presented here investigate to what extent the presence of a large fundamental affects the recovery of harmonic amplitudes and phases.

II. Theoretical Considerations

Quantisation by the ADC adds error to the sampled values. If the input signal is not correlated with the quantisation levels of the ADC, the quantisation error is effectively random

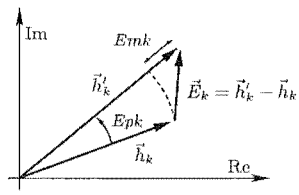


Fig. 2. The error vector of the quantisation, $\vec{E}_k$, as difference between the measured vector $\vec{h}'_k$ of magnitude and phase, and the actual vector $\vec{h}_k$. Em_k is the difference of measured and actual harmonic magnitudes $Em_k = |\vec{h}'_k| - |\vec{h}_k|$, and Ep_k is the difference of harmonic phases $Ep_k = \angle\vec{h}'_k - \angle\vec{h}_k$. k is the harmonic order.

and uncorrelated, and behaves like additive noise. The input signal is uncorrelated with the quantisation levels if it is spectrally dense, that is, if a number of frequencies in the signal have amplitudes larger than the quantisation interval of the ADC. Also, any signal can be made to be uncorrelated to the quantisation levels by dithering the signal [3].

If the ADC has an ideal transfer function with a quantisation width of w, the quantisation error is uniformly distributed in the interval $[-\frac{w}{2}; \frac{w}{2}]$, and has zero mean and variance $\frac{w^2}{12}$. The quantisation noise power q is equal to the variance of the quantisation error

$$q = \frac{w^2}{12} \tag{1}$$

Since the quantisation errors in each signal sample are effectively random, the quantisation noise in the spectral domain is complex Gaussian, i.e. its real and imaginary parts are each Gaussian distributed with zero mean, and are statistically independent of each other [4]. This is so because each spectral output of the FFT is a weighted sum of the signal samples input to the FFT, so the central limit theorem applies to the FFT outputs [5].

Since the quantisation errors in each signal sample are uncorrelated with each other, the quantisation noise in the spectral domain is white, i.e. the quantisation noise power q is spread equally across all frequencies, including DC. For an N-point FFT, the noise power in each output "bin" of the FFT is $\frac{q}{N}$. The FFT output bins represent the magnitudes and phases of a complex-exponential decomposition of the input signal. The nth harmonic in the input signal appears in both positive-frequency and negative-frequency bins in the FFT output, with the amplitude in each bin equal to half the amplitude of the harmonic, as described by the trigonometric identity

$$\cos x = \frac{1}{2}(e^{jx} + e^{-jx}) \tag{2}$$

Therefore, when using an FFT for harmonic analysis, harmonic amplitudes are measured by taking twice the value of the positive frequency bins of the FFT output, and the noise power or variance for each harmonic except DC is

$$Q = \frac{4q}{N} \tag{3}$$

Each harmonic component of the quantised signal can be described as a vector with an associated length (its magnitude) and angle (its phase). The diagram in figure 2 shows in

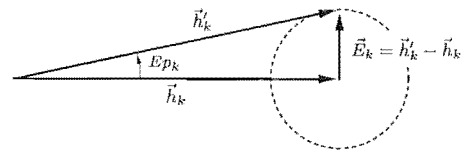


Fig. 3. The trigonometry underlying the phase error Ep_k, to demonstrate the relationship of phase error variance to harmonic magnitude $|\vec{h}_k|$.

cartesian form the true value $\vec{h}_k$, the measured value $\vec{h}'_k$ arising from the quantised signal, and the measurement error

$$\vec{E}_k = \vec{h}'_k - \vec{h}_k \tag{4}$$

for each harmonic order k. The measurement results a harmonic analyser reports to its user are the magnitude and phase of $\vec{h}'_k$.

The statistics of the noise in the measured harmonics are those of the error vector $\vec{E}_k$, which as previously noted has a complex Gaussian distribution. The statistical properties of $\vec{E}_k$ are the same for all k, except $k = 0$ (DC). Its magnitude has a Rayleigh PDF (probability density function) with variance Q, and its phase has a uniform PDF over the whole circle [4]. The measured harmonic value $\vec{h}'_k = \vec{h}_k + \vec{E}_k$ is complex Gaussian, plus a constant vector. Its magnitude has a Rician PDF [4].

Of interest to the user are the magnitude error

$$Em_k = |\vec{h}'_k| - |\vec{h}_k| \tag{5}$$

and phase error

$$Ep_k = \angle\vec{h}'_k - \angle\vec{h}_k \tag{6}$$

The magnitude error Em_k has a distribution of Rician form, but with the origin shifted by $|\vec{h}_k|$. For large $|\vec{h}_k|$ (harmonic power $\gg$ noise power, $|\vec{h}_k| \gg |\vec{E}_k|$) the distribution of Em_k becomes approximately Gaussian with a mean of zero and a variance of $\frac{Q}{2}$. For small $|\vec{h}_k|$ (not much larger than the noise power), the form of the distribution becomes skewed and the mean significantly larger than zero.

The trigonometry underlying the phase error Ep_k is shown in figure 3. In the right-angled triangle shown,

$$\sin(Ep_k) = \frac{|\vec{E}_k|}{|\vec{h}'_k|} \tag{7}$$

When the spectral phase error Ep_k is small, i.e. $|\vec{E}_k| \ll |\vec{h}_k|$, $\sin(Ep_k) \approx Ep_k$ and $\vec{h}'_k \approx \vec{h}_k$:

$$Ep_k \approx \frac{|\vec{E}_k|}{|\vec{h}_k|} \tag{8}$$

The phase of the error vector is uniformly random over the full circle (as indicated by dashed circle in figure 3), but since the error vector phase is statistically independent of the error vector magnitude, the variance of Ep_k remains approximately proportional to the variance of $|\vec{E}_k|$ and inversely proportional to the true harmonic magnitude $|\vec{h}_k|$, for small Ep_k (i.e. high signal-to-noise ratio).

From the foregoing discussion, the following proportionalities exist for the magnitude and phase errors relative to the

TABLE I
PROPORTIONALITIES FOR MAGNITUDE AND PHASE ERRORS.

Dependence on	Magnitude error variance	Magnitude error std dev		Phase error variance	Phase error std dev	
Harmonic order k	independent		(9)	independent		(10)
Harmonic amplitude ($\lvert\vec{h}_k\rvert$) ($SNR \gg 1$)	independent		(11)	$\frac{1}{\lvert\vec{h}_k\rvert^2}$	$\frac{1}{\lvert\vec{h}_k\rvert}$	(12)
Number of quantisation levels (L)	$\frac{1}{L^2}$	$\frac{1}{L}$	(13)	$\frac{1}{L^2}$	$\frac{1}{L}$	(14)
FFT length (N)	$\frac{1}{N}$	$\frac{1}{\sqrt{N}}$	(15)	$\frac{1}{N}$	$\frac{1}{\sqrt{N}}$	(16)

transform (FFT) length N, the number of quantisation levels L, and the amplitude $|\vec{h}_k|$ of the harmonic with order k.

III. SIMULATION

A. Model

A system to generate and analyse harmonics is modelled as shown in figure 4. Because the focus for these simulations is on the ADC, analog input circuitry like scalers and low-pass filters are not represented in the simulation. Their contributions to the frequency and phase errors are constant and can be compensated for in the spectrum.

As the first stage in the simulation, a periodic signal $h(f)$ is specified in the frequency domain. It is specified by an amplitude A_0 at a frequency of 0 Hz (DC), and amplitudes A_k and phases γ_k of the fundamental frequency and its harmonic frequencies up to a maximum harmonic H. The time-domain signal $s(t)$ corresponding to $h(f)$ has the formula

$$s(t) = A_0 + \sum_{k=1}^{H} A_k \sin(kt - \gamma_k) \tag{17}$$

with H as the highest order harmonic present in the signal. The simulation creates samples of $s(t)$ at regularly spaced times,

$$samples = \sum_{n=0}^{N-1} s(n \cdot \frac{360°}{N}) \tag{18}$$

where n is the sample number, and N is the number of points per period T of the fundamental frequency. $\frac{N}{T}$ is the sampling rate. The sample points of $s(t)$ are evenly spaced by $\frac{1}{N}$, with the first point at $t = 0$, and the last point a distance of $\frac{T}{N}$ before $t = T$, or at $t = \frac{N-1}{N}T$. Repeatedly concatenating the sets of sample points results in a continuous signal.

To simulate the effects of quantisation by an analog-to-digital converter, a quantisation function is used which rounds intervals in the input to the same output value. The number of quantisation steps is determined by the width of the ADC in bits. For a width of B bits, the number of steps L is $L = 2^B - 1$, and the width and height of each step are $w = \frac{2}{2^B-2}$. To simplify the simulation, it is desirable to keep the range of the input amplitude symmetric around zero. With an even number of steps and one step for the centre, one step remains which cannot be used by a symmetric transfer function. The function as used is odd-valued, and implemented with an input range of ± 1. All amplitudes are discussed with respect to ± 1 (unity).

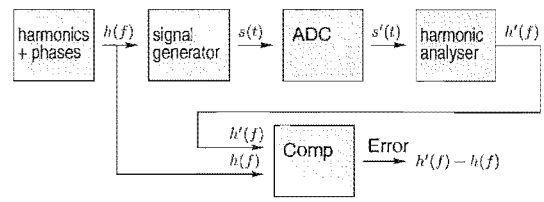


Fig. 4. Block diagram of the simulation model.

Within the scope of these simulations, no further digital filtering is performed.

The result of the quantisation is then processed with a harmonic analyser, based on a fast Fourier transform. The harmonic analyser function returns the magnitudes and phases of the discrete Fourier spectrum of the signal being analysed, such that when analysing a signal $s(t)$ with no quantisation noise, it returns the true harmonic magnitudes and phases, $\vec{h}'_k(s(\vec{h}_k)) = \vec{h}_k$ (see figure 4). The output of the harmonic analyser with the quantised signal $s'(t)$ is then compared with the true harmonic values which were used to synthesise the input signal.

B. Simulation Parameters

The simulations are performed with Octave [6]. The model described in the previous section is run with a range of values, in order to examine the magnitude and phase errors in relation to

1) harmonic magnitude and ADC width at a fixed harmonic order,
2) harmonic magnitude and harmonic order at a fixed ADC width, and
3) FFT length and ADC width at a fixed harmonic order.

The effects of ADC width and harmonic order are examined over several bands of harmonic amplitude relative to the amplitude of the fundamental. Each band is given by a lower and an upper amplitude limit. This banding is necessary because the harmonic phase error is inversely proportional to the harmonic amplitude. Phase errors of harmonic amplitudes near zero would otherwise dominate the results and conceal the relationship between harmonic amplitude and phase error.

Because the ADC input range is set to ± 1, care has to be taken that the signal does not exceed this limit. The DC value is set to 0. The amplitude of the fundamental is set to 0.5 (i.e. the fundamental is given by $0.5 \cdot \sin(t)$), and giving the harmonics random amplitudes with an upper limit of 0.025 almost always produces signals with peak-to-peak amplitudes $\leq \pm 1$. Therefore, the fundamental is at least 40 times larger than the largest harmonic. The effect of harmonic amplitudes on the resulting error is examined by giving the harmonics random amplitudes within bands. Bands are set in decade intervals starting with 0.025 and are one interval wide, i.e. bands are 0.025 to 0.0025, 0.0025 to 0.00025, and so forth. Each band is examined separately, and for each band each harmonic is independently given an amplitude randomly generated with uniform distribution within the limits of the band.

The sampling rate is specified to the simulating program as the number of samples per period of the fundamental frequency. The number of harmonics is set to 128 (including DC), a higher number than is expected to be of interest in real systems. Therefore the number of sample points is set to 256 to satisfy the requirements of the sampling theorem. The Fourier transform length is equal to the number of samples.

Because of the finite numerical precision of the machine's arithmetic when calculating the FFT, the output of the FFT contains arithmetic noise. Harmonics whos magnitude is comparable to or less than this arithmetic noise level will be swamped by the arithmetic noise. The reported phase values of these noise-harmonics are meaningless and effectively random over the full circle. Because the phases of the harmonic analysis are of particular importance and arbitrary values make a comparison of the phases of input and result difficult or impossible, harmonic amplitudes of 10^{-9} iustead of 0 are used. This is sufficiently small to not affect results, and sufficiently large compared to the machine precision of approximately 10^{-16} (64 bit floats).

For each ADC width, harmonic order, or FFT length under consideration, the process of creating a harmonic signal, transforming it, and comparing the result with the initial value is repeated 200 times. For the magnitude and the phase of the error, the mean and standard deviation of these 200 results are calculated and plotted against the variable under consideration.

IV. Results and Discussion

The results of the harmonic magnitude and phase errors as a function of harmonic order at a fixed ADC width are given in figures 5 and 6 for the magnitudes and in figure 7 for the phases. These graphs show that the error is independent of harmonic order, which is consistent with the theory. None of the proportionalities in table I contains the harmonic order.

Mean and standard deviation of the magnitude error and mean of the phase error are independent of the ratio of the harmonic magnitude to fundamental magnitude, but only down to a certain limit. The particular limit is related to the quantisation width w of the ADC, or ADC width. When the harmonic magnitude is well above the noise, the mean of the error is essentially zero.

For magnitudes close to or below the quantisation noise level, the mean increases. This is understandable because magnitudes can not be negative. Adding random noise values with a mean of 0 to positive values that are smaller or 0 can not result in sums with a mean of 0. This is then reflected in the standard deviation as well as a non-zero mean.

The standard deviation of the phase error decreases linearly with decreasing harmonic amplitude (figure 7), i.e. the standard deviation of the phase error is inversely proportional to the harmonic magnitude. This is in agreement with equation 12. Again this is true only up to the limit of noise amplitude. The 2.5×10^{-5} line is at 60°, not at 100°.

Another possible contributing factor to the non-zero mean of small harmonic amplitudes is that when the harmonic amplitudes are less than the quantisation intervals of the ADC, the samples may no longer be uncorrelated with the quantisation levels, in which case the quantisation error is no longer random. Any such correlations can be suppressed by dithering the input signal, as mentioned in section II.

TABLE II
HYPOTHETICAL THD VALUES

A_1	0.5	0.5	0.5	0.5	0.5	0.5
A_n each	0.05	0.025	0.01	0.0025	0.001	0.00025
$2 \leq n \leq 6$, THD	31.6%	15.8%	6.32%	1.58%	0.63%	0.16%
$2 \leq n \leq 11$, THD	22.4%	11.2%	4.47%	1.12%	0.45%	0.11%

In any case, harmonics with magnitudes this low in comparison to the fundamental are unlikely to be of practical interest in a power distribution system, because their effect on power quality is negligible.

Because figures 5, 6 and 7 show that the spectral errors are independent of harmonic order, the simulation of the effect of varying ADC width is computed only at a single harmonic order. The 5th harmonic is chosen because it is commonly of interest. All other settings remained unchanged.

The results of the harmonic magnitude and phase errors as a function of ADC width at a single harmonic order and at different harmonic magnitude levels are given in figures 8 and 9 for the magnitudes and in figure 10 for the phases. The mean of the magnitude error decreases with increasing ADC width and is zero for sufficiently large harmonic amplitudes. The standard deviation decreases linearly by a factor of 2 per 1 bit increase in ADC width (doubling of the number of quantisation steps), which is in agreement with the theory (equation 13).

The mean of the harmonic phase errors is 0 for sufficiently large harmonic amplitudes. The standard deviation decreases by a factor of 2 per 1 bit increase in ADC width, which again agrees with the theory (equation 14).

Both magnitude and phase rms error are directly proportional to the size of the quantisation interval, or the inverse of the number of quantisation steps. Both are also proportional to the inverse of the square root of the FFT length. Both factors are intuitively understandable. The rms error in the signal is proportional to the size of the quantisation step, and the total error power is equally divided between each harmonic.

Figure 10 is the most useful one for evaluating the phase recovery performance of a power quality monitor. For example, a system with an 8 bit ADC and 128 point FFT can resolve harmonic phases of unit amplitudes between 0.0025 and 0.025 with a standard deviation of 1.5°, and of amplitudes between 0.00025 and 0.0025 of 40°. A system with a 12 bit ADC has a standard deviation of 0.1° and 2° respectively. (The fundamental has a unit amplitude of 0.5.)

These amplitude values can be related to the total harmonic distortion, THD, which is used as a measure for the overall harmonic content in a signal. It is defined as

$$THD = \frac{1}{A_1} \cdot \sqrt{\sum_{n=2}^{\infty} A_n^2} \tag{19}$$

and usually given as a percentage. The THD can be used as a measure for the line losses due to non-linear loads. With an FFT length of 128, up to the 63rd harmonic can be recovered.

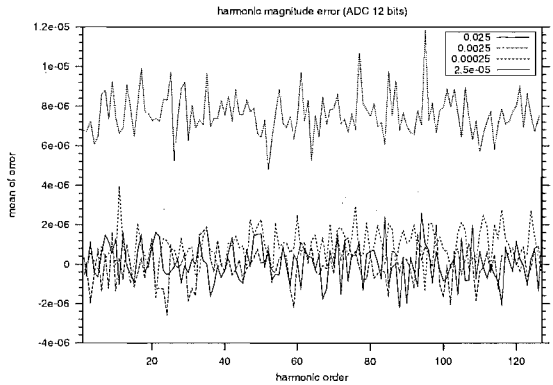


Fig. 5. The mean of magnitude errors is independent of harmonic order.

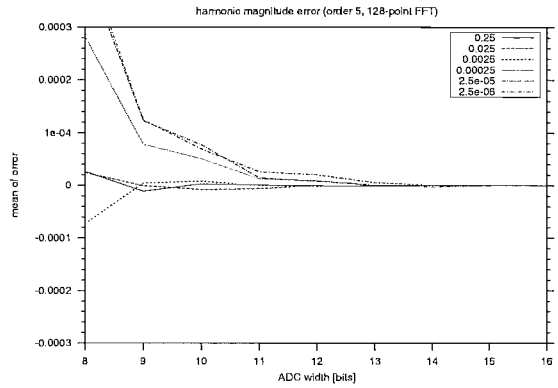


Fig. 8. Harmonic magnitude errors decrease with increasing ADC width.

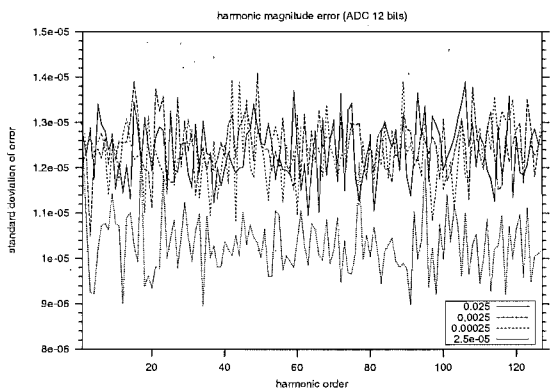


Fig. 6. The standard deviation of magnitude errors is independent of harmonic order.

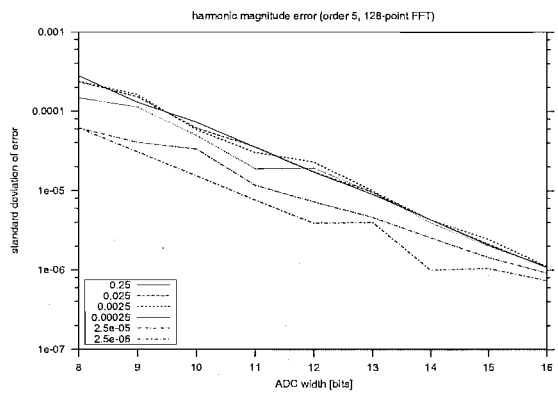


Fig. 9. Harmonic magnitude errors decrease with increasing ADC width.

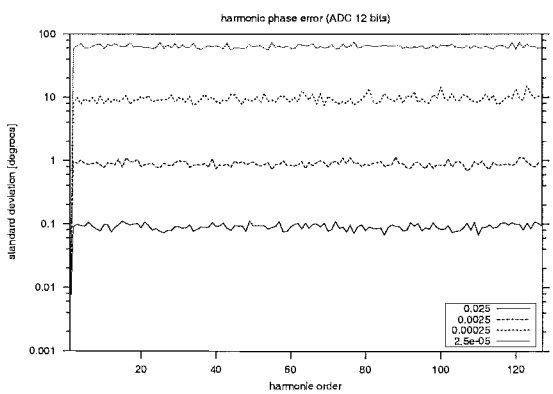


Fig. 7. The standard deviation of phase errors is independent of harmonic order, and is inversely proportional to harmonic magnitnde.

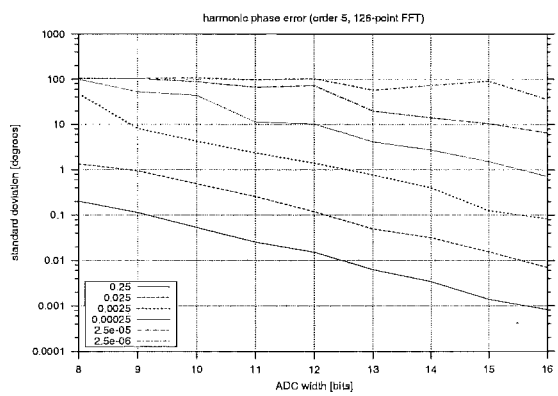


Fig. 10. The standard deviation of the phase errors decreases with larger ADC widths.

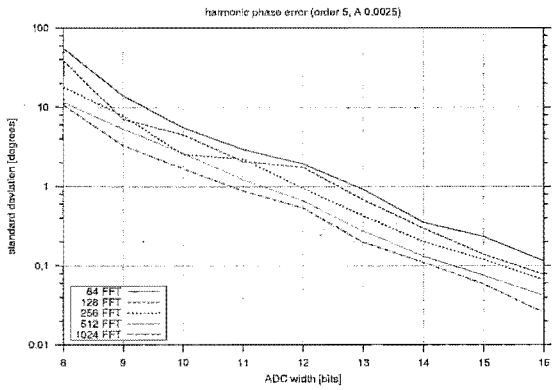


Fig. 11. The standard deviation of the phase error decreases with larger ADCs and longer transforms.

As an example, table II shows the THD values that would result from a system with the fundamental amplitude $A_1 = 0.5$ and either 5 or 10 harmonics with an equal amplitude A_n and the remaining harmonics not present.

As discussed in section II, the phase noise power is proportional to the inverse of the transform length, therefore the phase noise standard deviation is proportional to $\frac{1}{\sqrt{N}}$. Figure 11 shows the phase noise standard deviation as function of ADC width for various FFT lengths between 64 and 1024, and for two different harmonic magnitude bands. The distance between the lines is approximately constant and equal to $\sqrt{2}$, which agrees with equation 16.

An increase in FFT length implies a corresponding increase in sample rate. When designing an acquisition system, this can be used to decide on a tradeoff between ADC width, and sample rate and computing power [2]. The computational cost of an FFT is proportional to $N \log N$.

The simple quantisation by the simulated ADC produces a flat distribution of noise over the spectral range. Certain types of ADC exist which have a noise spectrum weighted towards higher frequencies, with a 6 dB/octave increase [7]. When used with oversampling, these ADCs push most of the quantisation noise to higher frequencies and out of the frequency range of interest, giving harmonic noise power approximately proportional to $\frac{1}{N^2}$, which is substantially lower than the $\frac{1}{N}$ from simple quantisation.

Making use of oversampling is also expected to increase accuracy, but was not investigated within the scope of these simulations. The effects of oversampling on magnitude and phase noise are equivalent to an increase in transform length. Oversampling averages R number of points into each new value, reducing quantisation noise variance by a factor of R, or rms noise by $\sqrt{R}$. Amplifier and ADC non-linearities result in spectral errors whose amplitude is not affected by the oversampling ratio.

V. Conclusions

The effects of quantisation noise on the ability to recover phase information have been examined, and simulated with different parameters. The error for recovering phase information is determined by the quantisation noise introduced by the analog-to-digital conversion. The simulations confirm that the phase error is independent of harmonic order. The magnitude error is also independent of harmonic order, and is independent of the magnitude of that harmonic, for magnitudes significantly larger than the quantisation noise magnitude. The rms phase error is inversely proportional to the harmonic magnitude.

Both magnitude and phase rms error are inversely proportional to the ADC width and the square root of the FFT length.

These two relationships are the key factors for assessing the capabilities of a data acquisition system. To increase the precision of the instrument, either or both of the ADC width or the transform length (and therefore the sample rate) can be increased. The graphs in figures 10 and 11 indicate the relationships.

The exact requirements of a data acquisition system depend on the particular application, but a system with 16 bit ADC appears unlikely to be required for power quality monitoring. A 12 bit ADC gives phase errors in the order of a few degrees, and an 8 bit ADC in the order of some tens of degrees, for harmonic magnitudes likely to be of interest for electricity supply applications.

Acknowledgment

The authors wish to acknowledge the support given by Transpower New Zealand Ltd in the early phases of this research activity.

References

[1] V. Kuhlmann, C. P. Arnold, and M. B. Dewe, "System requirements of real time continuous data acquisition for power quality monitoring," 2005.

[2] V. Kuhlmann, "Synchronous measurements and power quality monitoring," Ph.D. dissertation, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Submitted 2004.

[3] J. Vanderkooy and S. P. Lipshitz, "Resolution below the least significant bit in digital systems with dither," *Journal of the Audio Engineering Society*, vol. 32, no. 3, pp. 106–113, Mar. 1984, correction in 32(11) p. 889 (Nov 1984).

[4] A. B. Carlson, *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*, 3rd ed. New York: McGraw-Hill, 1986.

[5] R. N. Bracewell, *The Fourier Transform & Its Applications*, 2nd ed. McGraw-Hill, 1978.

[6] http://www.octave.org/ Octave is an open-source programming language for numerical computations, which uses matrix-like data structures.

[7] J. Watkinson, *The Art of Digital Audio*, 3rd ed. London: Focal Press, Dec. 2000.

Volker Kuhlmann undertook his undergraduate studies at the University of Siegen, Germany, and gained his M.E. at the University of Canterbury, New Zealand. He is currently completing his Ph.D.

Alastair Sinton is a commercial software developer with interests in signal and image processing. He gained Ph.D. and B.E. in electrical engineering at the University of Canterbury, New Zealand, in 1986 and 1982. He is a member of IEEE and ACM.

Mike Dewe was Senior Lecturer at the University of Witwatersrand in Johannesburg, South Africa, before joining the University of Canterbury in 1978. He was appointed IT Director in 2000, after having been Associate Professor in Electrical and Electronic Engineering. He also has a total of 12 years industry experience.

Chris Arnold is a Senior Lecturer in Electrical and Computer Engineering at the University of Canterbury, New Zealand. He is a Chartered Engineer and a Member of the IEE. His main interest is in power system dynamics.

A.7 System Requirements Paper

Submitted to the Elsevier Electric Power Systems Research [109].

System Requirements of Real Time Continuous Data Acquisition for Power Quality Monitoring

V. Kuhlmann, C. P. Arnold, M. B. Dewe
University of Canterbury, Christchurch, New Zealand
volkerkuhlmann@gmx.de

Abstract—**Data acquisition systems for power quality monitoring have complex requirements which also vary with the intended application. Overall, the complexity of an instrumentation system for power system monitoring is most dependent on the number of channels and sites which must be analysed, and the required time stamping accuracy. Three broad categories of system configurations are described with their basic parameters. The emphasis is on a discussion of hardware and software issues which are involved in a system specification.**

Index Terms—**Power quality, data acquisition, power system monitoring, power system measurements, monitoring, power transmission control**

I. Introduction

WHEN faced with the task of monitoring or examining an electricity supply system, suitable instrumentation must be obtained. The requirements for a particular power quality monitoring system depend to a large extent on the application and the specific answers which are sought in the respective situation. This paper discusses the significant parameters for such systems, typical values for different types of application, and other relevant issues. A detailed ordered checklist [1] can be used as a template for generating a specific requirement specification for a power quality monitoring instrument.

Systems for power quality monitoring can be broadly divided into three areas of application with different complexity:

1) Equipment to measure a single 3-phase system at a single site.
   Such equipment consists of typically 3–8 channels. Time stamping of samples is relatively straightforward because only a single instrument is involved and high time stamping accuracy is not necessary.
2) Equipment to measure multiple 3-phase systems at a single site, such as a substation.
   Such equipment can consist of tens of input channels, and sampling of all channels must be synchronised if a high degree of coherency is required (for example when comparing harmonic phases). High coherency implies that all channels share a common sampling clock. If channels are not located within the same instrument enclosure, the sampling clock needs to be distributed. A high time stamping accuracy is not necessary.
3) Equipment to measure at more than one site, with each site being in one of the two configurations above.
   The sampling clocks of all involved instruments must be synchronised to a high degree if sample data is to be comparable. In practice this means synchronisation to a highly accurate common time source. This same method of synchronisation is also a possible solntion for implementing coherent sampling in the second case above when multiple enclosures are involved.

For each of the above categories, the duration of measurements can vary widely. Examples of short-term applications include confirming a specific operational status, and using a mobile instrument to investigate a specific problem that has started to occur. They are typically characterised by event-driven recording. An example of a permanent application is continuously monitoring one or more characteristics of the electricity supply network for ongoing conformity, using a fixed installation of instruments.

The characteristics which can be examined depend on both the hardware and the software of the monitor. The hardware defines a permanent maximum limit for the rate at which sample data can be acquired, whereas the software defines the type and method of analysis that is performed on the data. By modifying the software, the monitoring system can be adapted to new and/or specific tasks and new standards for calculating power quality. Generous dimensioning of the hardware allows for a higher level of flexibility with respect to future analysis requirements.

Implications of the above three configurations and other factors present in power quality applications are discussed in more detail in this paper. A particular issue is the storage and analysis of prodigious quantities of data, and the importance of reducing this to relevant subsets.

II. System Aspects

The requirements with the largest impact on overall system design are the need for highly accurate time stamping, continuous operation, and noise issues.

For highly accurate time stamping, an off-the-shelf system is not readily available, although it may be possible to build a system from off-the-shelf components. Design choices made for the details of sampling, and the time stamping of those samples, have significant ramifications for the broad system layout. The use of buffers is difficult because the temporal relationship between sampling clock and samples might be lost.

The requirement of continuous operation precludes the use of buffers to extend the maximum sampling time. Data volumes can become high in a short time. Data must be

processed immediately to allow for continuous operation. It is essential that this processing causes a data reduction of several orders of magnitnde, resulting in a volume of data which can realistically be stored. Data reductions can be achieved by identifying regions of interest, by transforming the time domain data into the frequency domain, and by not storing the FFT result for every mains cycle. Keeping one FFT for every mains cycle is unlikely to be of interest in normal circumstances.

The location of ADCs with respect to the signal source is critical in an electrically noisy environment such as a switchyard or industrial plant. Shielded cabling is not necessarily adequate, and the electrical characteristics of the sensor, the cable, and the ADC input have to be carefully matched. This can be difficult in practice. Where accuracy is most important, a solution is to locate the ADCs physically close enough to the sensors to prevent significant noise contamination. This generally means that ADCs have to be located in separate enclosures if the signal sources are further apart than a few metres. Multiple enclosures have major implications for coherent sampling.

Other factors that must be considered for overall system design are the number of input channels, the bus system used, and its maximum card size. The maximum card size determines how many ADCs can be integrated on one card. This has cost implications because in any system design there is a fixed cost associated with each card. The bus system determines the maximum number of cards. The overall maximum number of channels determines the kind of measurement that can be undertaken before a sudden increase in cost, necessitated by a second system, takes place. However, if a limited number of signal sources are in close enough proximity, the maximum number of channels may be of lesser importance.

For fixed installations, the instrument should be supplied with mains power. For portability and temporary setups, batteries are a good option, although their overall cost (weight, volume, charging, casing) and inconvenience can be significant. Battery life must be sufficient for the required measurement duration. The use of batteries may be the only option when floating the whole system at high potential or for avoiding ground loop issues.

The data acquisition units should be networked with another computer for user interface, control, and data storage / backup. It does not make sense to design even a simple system without networking capabilities. A data link allows remote instruments to be controlled, and data to be retrieved from them. It is possible to make the network an integral part of the instrument setup, by sending all sample data to a fast central computer for processing. If the system is to interoperate with other monitor and control equipment, suitable interfaces need to be provisioned. If the system is designed for central data processing, sufficient network bandwidth must be available. If basic filtering is performed on the ADC units and final processing on a central computer, network bandwidth requirements can drop to less than 20 kbyte/s/channel.

This central computer could also perform time keeping, with the time being transferred to all ADC systems for time stamping, removing the need for individual external time sources at each ADC unit. However, highly accurate time transfers can not currently be achieved over Ethernet [2].

For mains-synchronised sampling, the mains frequency needs to be determined, for example by determining the cycle time of a mains-derived signal connected to a suitable interrupt with a low-pass filter. This can happen on all ADC units individually, or on the central computer followed by transfer over the network to the ADC units. It may be easier to obtain the mains-derived signal at the central computer, depending on the power supply situation of the ADC units. However, doing this centrally introduces measurement uncertainties which need to be considered.

If all processing is to be done by the instrument, then clearly the CPU must be powerful enough to deal with all the design requirements. However, it is expedient to oversize the CPU to allow for future modifications.

Depending on the particular application or intended range of uses, digital inputs and outputs can be considered. If the digital inputs are time-stamped with the same mechanism as the analog inputs, event capture channels are available for interfacing to other equipment. Digital outputs can be used for control and signalling functions.

Some applications will require that some or all inputs and outputs are galvanically isolated, for example if the whole instrument is floated. Galvanic isolation is also advantageous for avoiding ground loop problems, which would be a safer design for a more general-purpose instrumentation system. Isolating digital I/O lines is relatively straightforward. The components for isolating analog inputs introduce their own frequency and phase response, and are costly. Similarly, overvoltage protection for inputs to (for example) 250 V_{ac} makes a system more robust against accidental misuse, but the costs have to be considered.

As mentioned previously, a commercial system which fulfils all the hardware requirements to the highest level is unlikely to be available. Analysis software tailored to typical applications is not currently available either, irrespective of low or high end systems. A system should be configured from commercially available components wherever possible to reduce cost and development time, especially for a low number of systems. As a general rule, the higher the requirements are, the more hardware will have to be specifically designed for the application. Tradeoffs should be considered. For example, if an ADC card is commercially available for all but the desired input voltage range, it would be more economic to use this card with a pre-scaler in a small external case than to design a new ADC card.

Tradeoffs exist between hardware and software implementation. Larger oversampling factors or transform lengths increase resolution and the computational cost, a wider ADC increases resolution and the hardware cost. The frequency of the system under investigation can be determined by computation with the measured values, or by using zero-crossing circuitry in connection with a counter.

A number of environmental factors need also be considered. Operation outdoors, unless short term in fine weather, requires protection class IP64 for the equipment, whereas standard desktop PC cases are sufficient for indoors. Rugged

casing makes transport easier, but is unnecessary for fixed installations. Adequate electromagnetic shielding is a design requirement for any instrumentation system. In extreme cases, a minimum distance to strong magnetic field sources may have to be kept.

Relevant government and industry regulations covering aspects such as electrical safety or data exchange need to be taken into account.

A system which compensates for sensor and converter characteristics when reporting values is easier and faster to use. This may require automatic or semi-automatic calibration facilities.

The required level of instrument reliability, which depends on the intended application, affects the overall system design. This should be considered right from the start of the design.

III. Input Stage

The input stage of general-purpose instruments needs to accommodate a wide range of source signal types. This is less important for more specific or fixed installations.

The design of the analog front-end and the ADC stage itself is critical with respect to the level of environmental noise resistance. Any design faults here reduce the effective ADC width when the lowest order bits of the ADC fall below the noise floor. This is also especially significant for the connection from the signal source to the ADC input. Appropriate shielding is necessary, especially in a noisy switch yard environment.

The range of input voltage of any instrumentation system is generally fixed, and tends to be in the order of $\pm 1\,\mathrm{V}$ to $\pm 10\,\mathrm{V}$ for commercially available data acquisition systems.

In power systems, voltage transformers (VT) typically are designed to output 50 V or 110 V at nominal system voltage. Current transformers (CT) are typically designed to output 0.5 A, 1 A, or 5 A at nominal maximum system current. Their output voltage is therefore also dependent on the load. The current can be monitored by measuring the voltage developed across a small resistor inserted in series with other instruments. The total VA burden applied to the CT must not exceed its design rating. Care must be taken to ensure that the CT secondary circuit is never open-circuited as dangerously high voltages will occur. Similarly a VT's secondary circuit must not be overrated and must never be short-circuited as a dangerously large current will flow.

The characteristics of CTs and VTs have been optimised for power system frequency. The output of harmonic frequency signals will suffer some degredation in both amplitude and phase. For the most accurate measurements, the characteristics should therefore be obtained. If saturation is reached, compensation is not possible. A system which is able to auto-correct for sensor characteristics, or is able to at least establish the sensor characteristics automatically with simple manual interaction, is user-friendly. However, while this may work well for equipment like current clips, the scope for characterising CTs or VTs already in service is more limited.

The large voltage range which has to be accommodated has two consequences: external voltage dividers / pre-scalers have to be used with industry-standard data acquisition equipment, and appropriate isolation to mains voltage standard must be provided, in accordance with respective safety regulations.

Automatic gain adjustment (auto-ranging) can increase the effective dynamic range of the ADC, but has the following disadvantages:

1) The switching is not instantaneous, therefore some period of time is not captured adequately. The sample time stamping needs to be able to cope with this. The main factor of the switching delay is the reaction time of the software.
   If transient capture is a part of the system's objectives, switching must be implemented in a way that does not interfere significantly with the transient waveform.
2) If suitable solid state switches can be sourced, the input range needs to be switched in sync with the sampling.
3) The analog circuitry required to implement this introduces further distortions and needs careful shielding to minimise the pickup of environmental noise.
4) Adjustable input ratios are not usually provided with commercially available ADC cards, or at least not with the desirable scale factors. If a power monitoring system can be otherwise implemented completely with commercially available components, adding a pre-scaler as custom-built hardware adds difficulties to the design.

Commercial systems are specified with respect to their performance for signals connected at their inputs. It can not be assumed that the signal at this point is a correct representation of the quantity to be measured, because the transfer characteristics of the sensor (transducer, current probe, etc.) must be taken into account. If these characteristics are known, they can be compensated for by the data processing software.

IV. Sampling and Sample Clock Generation

Two of the key parameters of any data acquisition system are the sampling rate (frequency) and the effective size of each sample when digitised (bits). The minimum sampling rate depends on the highest order harmonic that is considered to be of interest, and the degree of accuracy to which the harmonic phase information is desired. The relationship between sampling rate, ADC width, and obtainable harmonic phase is examined elsewhere [3].

To recover harmonic magnitudes, the minimum sampling rate is twice the frequency of the highest-order harmonic to be recovered, for example 5 kHz for the 50th harmonic of a 50 Hz system. In practice, choosing a slightly higher value would avoid any marginal error. Sampling rates in this range are sufficient to calculate total harmonic distortions (THD).

With oversampling, the design of any analog low-pass anti-aliasing filter for the ADC input is simplified. The main advantage is that oversampling, in connection with a digital FIR (finite impulse response) filter, reduces quantisation noise introduced by the ADC. A higher rate of oversampling has the potential to increase the effective ADC resolution (number of bits), but also increases the bandwidth and FIR filter computational needs. 8× oversampling should not present

great difficulties in terms of additional computational requirements for the FIR filter at these comparatively low sampling frequencies.

ADC widths of 12 bits are widely available. Widths of 8 bits are common for very high sampling rates in the megahertz range, these high rates offer no advantage for power quality instrumentation. Widths of 16 bits are also commonly available. Widths in excess of 16 bits are relatively uncommon because noise interference problems become difficult to control, and are not likely to be cost effective. 12 bits provides a good compromise, while 16 bits will satisfy the most stringent requirements [3].

The digital signal processing computational requirements are unaffected by the ADC width, because the minimum data type which can be handled is 4 bytes, assuming the calculations are performed in floating point format. It would be possible to perform calculations with sufficient accuracy with integer arithmetic only. 16 bit integers would be sufficient for 8 bit ADCs, however 12 bit ADCs would require 32 bit integers. The relatively small advantage in execution speed of 32 bit integer over floating point numbers in modern processors needs to be weighed against the time and effort of having to software engineer all computations in integer.

A certain amount of control over the sampling clock generation is desirable. Commercial systems always allow programming of the sample rate directly, based on a crystal oscillator. Supplying the sampling clock externally is also a standard feature. The latter is necessitated by the need to sample at identical frequencies across a number of channels, as well as by sampling with a known phase relationship across channels. This becomes an issue as soon as channels are spread over more than one data acquisition printed circuit card.

For some applications, it is important that the sampling clock skew between channels is kept to a minimum. This includes applications for harmonic state estimation or harmonic power flow, where accurate harmonic phase information must be obtained. It is also important for transient analysis. Using one of the ADC cards to generate the sampling clock, and supplying this as external signal to the other cards, is a standard feature of commercial ADC cards and solves this part of the clock skew problem. Clock skew is also created by multi-channel ADC cards which are multiplexed. These cards sample all channels sequentially. The clock skew introduced by multiplexing may however be deemed negligible if the total time taken to sample all channels is much smaller than the sampling period. If this is not the case, one ADC per channel should be opted for.

If the type of ADC requires a sample-and-hold (S&H) circuit, this is usually already integrated into the ADC and its error contribution included in the ADC's error specification.

Power quality applications, which involve the computation of Fourier transforms, usually specify a sampling rate which is a multiple of the fundamental frequency. This reduces issues in FFT calculations by rectangular windowing, and simplifies software design and processing needs. Sampling clock generation is therefore characterised by:

1) Samples per second
2) Sampling per an externally supplied clock signal

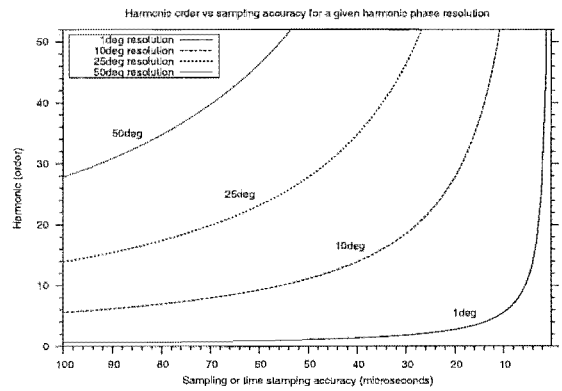


Fig. 1. The maximum harmonic order which can be resolved with a given timing accuracy and phase angle resolution. This graph is based on a 50 Hz system. For a 60 Hz system, divide the harmonic order by 1.2.

3) Samples per mains cycle

Sampling at a multiple of the fundamental frequency can be achieved by continually changing the programmed sample rate, if the actual mains frequency is known to the data acquisition system. However, the resolution with which the sampling rate can be programmed will need to be considered. It has to be decided whether it would be sufficient if the sampling rate could for example only be set in terms of whole samples per second.

V. Time Stamping

Time stamping, or time tagging, of sample data is of much interest. Two cases have to be distinguished: single-site and distributed measurements. If a single instrument with a number of channels is involved, coherency is generally inherent if the input channels are not multiplexed. The resolution of the time stamping is as provided by the system, and the accuracy is irrelevant because all samples can be related to each other. The situation is similar for multiple systems at the same site if they share a common sampling clock. One system is the time master for the others. The characteristics of the time transfer from the master system can not be neglected if the delay of the time transfer introduces an unacceptably high clock skew, but accuracy remains irrelevant.

For a distributed measurement setup, all systems need to be synchronised, but a time transfer from a master system is not practical or economic. The only practical solution is to synchronise all system to an external time source. The accuracy of this synchronisation to an external source, and therefore the accuracy of the time stamping, directly limits the resolution to which sample data from different measurement locations can be compared.

The accuracy of the sample time stamping translates into a limit for the degree to which harmonic phases can be compared. Figure 1 relates the harmonic order and the time which correlates to a given fraction (angle) of that harmonic. For example, 25° of the 14th harmonic correlates to 100 µs, whereas 1° of the 50th harmonic correlates to approximately

1 µs. The only practical and economic way to achieve a timing accuracy to this level is by utilising satellite time receivers.

Flexibility is increased if time stamping remains functional in the absence of a signal from the external time source. Time stamping of external events should be performed with the same accuracy as the time stamping of sample data. The standard IEC61850 [4] defines 5 different accuracy classes from 1 ms to 1 µs.

VI. Software

Individual analysis of particular sample data can be performed with general-purpose mathematical software like Matlab, Octave or Scilab, but this is not suitable for permanent unsupervised operation, or generating any kind of alerts. A higher level of automation is desirable. Possible functional requirements for the software therefore are:

1) Control of the beginning and end of sampling, both manual and automatic. A list of times when sampling starts and stops should be configurable, and this should include the possibility of repetition (for example every N seconds, hours, days or weeks). Activation or deactivation of sampling by an external signal may be required, especially for interaction with other equipment.
2) Methods of sample data processing. The instrument should be pre-loaded with a range of commonly-used filter and power quality analysis functions, including for example total harmonic distortion, flicker, dips and sags.
3) The ability to set thresholds and generate trigger conditions. Power quality parameters should be calculated according to applicable industry standards, and cause a trigger condition when preset thresholds are exceeded. This may require additional hardware or software interfaces to other equipment.
4) The ability to complement the existing built-in data processing functions by user-supplied functions, to allow for tailoring to specific conditions and problems. The user should only be concerned with programming the function itself, all other programming aspects should be of no concern.
5) The ability to configure storage of processed data for the short and long term, and to specify which data should be stored at all. It should be possible to specify a maximum storage size per result type, so that when this is reached, the oldest data is deleted first. Data which has not been downloaded by then and archived, whether this is automated or not, would be lost permanently. This should be organised in levels, comprising different time spans and levels of detail.
6) Compensation for sensor characteristics. For accurate measurements, it is necessary to compensate for the characteristics of transducers and probes. If this is desired, the system software needs to be able to deal with transfer characteristics supplied by the user in some suitable format.
7) Storage of time-domain data. If the ability to re-process the sample data at a later time is required, the sample data needs to be stored. Simplification of post-processing and reduction of data volume can be achieved by storing the data after it has passed noise and anti-aliasing filtering and sensor charateristic compensation (item 6). To limit resource requirements, a maximum duration for which time-domain data can be stored by the system should be specified.
8) Storage of data for a configurable time preceeding and following a trigger condition. Storage can be arranged in a similar way to that described in item 5. Whether the storage of time-domain data (item 7) is also needed has to be decided.
   This particular function is important and has large documentary and evidential value in case of serious faults like blackouts, when the cause needs to be established. Information about the state of the distribution network before the event is essential.
9) Remote control all the functions listed here via data network. TCP/IP is universally used for this and therefore guarantees interoperability with LANs.
10) Data reduction, as a general principle. It is impossible for anyone to work through the massive amounts of data which can be generated. Software must be capable of allowing the user to specify which information is important and which should be discarded.
11) Determination of mains frequency for controlling the sampling, replacing any equivalent hardware solution.
12) Encryption for control or data communication. This is important over shared networks to prevent unauthorised control.
13) Well-designed user interfaces, to aid efficient work flow. This is also true for control elements of the hardware.

Apart from functional requirements of the software, general aspects of development, implementation and support need to be considered. Multi-tasking capabilities aid in modularising the software and are a standard feature.

The choice of operating system (OS) has long-term implications. Aspects which need to be considered include long-term availability and support, and to what extent real-time features are required. Availability should not become a problem with any supplier with a large market share, and is implicit with open source software. Support by the supplier is not needed if support from third parties is available as well or instead. The ability to hire skilled staff should be considered. The more specialised the operating system, the more difficult this becomes.

If a real-time operating system for embedded systems is used, specialised tools are needed for development, and the total development time is likely to be longer, resulting in a higher cost. Average throughput can be relatively low. Using a general-purpose operating system results in a large number of low-cost development tools, and the software can for the most part be tested on any PC. A large amount of software is readily available and may be incorporated into the design (license issues permitting). A middle way between real-time and general purpose is a general purpose operating system with a real-time core.

The portability of software to different platforms is an advantage in heterogenous environments, especially when

hosts are networked. Wheu designing data storage formats, the compatibility of binary data with big-endian and little-endian processors is key to successful data interchange. Data exchange in text format such as XML has become popular for this reason, but the associated overheads (increased size and conversion time) have to be allowed for.

For the archiving of processed data a data base can be used, allowing for convenient access to past information. This adds a piece of complex software to the system, and another cost factor (unless a free data base is used). Equivalent functionality can probably be obtained by organising the storage of files in a specified way.

An instrument of this complexity calls for a large amouut of user documentation. This also applies to any custom-made hardware, and the system as a whole.

VII. Designing Custom Hardware

The development time and cost of designing custom hardware needs to be weighed carefully. Other tradeoffs have already been discussed.

It is more economic to design custom hardware to connect to an established interface or bus system. The interface should be chosen carefully because it will have long-term implications for future developments. Open bus systems which are supported by more than one manufacturer are preferable, this is true for both the hardware and the software running on that hardware.

Ideally, a platform for development of custom hardware would fulfil the following requirements:

1) be supported by multiple vendors
2) have sensibly-controlled specifications
3) have a sufficient throughput
4) have multi-master capabilities
5) offer a minimum card size for user hardware
6) offer a sufficient number of slots
7) allow a means to carry user-defined signals between user-designed cards
8) be maintainable (also true for software)

Any method of transferring time from one card to another through hardware would require some sort of bus system. The interconnecting bus systems can be difficult to implement physically. A convenient and tidy way of interconnecting cards is to use backplanes. A number of bus systems exist which provide unused lines for user applications.

If time stamping is implemented with the use of hardware counters, the full width of the counter can be transferred via a bus to the ADC cards. Alternatively, if only a few bus lines are available, the counter can be duplicated on the ADC cards, requiring only a clock and a reset line to be transferred by some means.

Provisioning a basic self-test which can quickly and easily verify the functionality of the hardware is advantagous for development, production testing, and field use.

VIII. Conclusions

The issues involved in selecting or designing an instrumentation system for power quality monitoring have been discussed for a wide range of possible requirements, and the constraints outlined.

Any requirement for time stamping with an accuracy aproaching 1 µs has a dominating effect on overall system topology. Other aspects with a strong influence on system design are noise immunity and physical deployability.

More stringent requirements for input levels, dynamic range, flexibility in sample clock generation, or highly accurate time stamping increase the likelihood that customised hardware is necessary, which noticably increases cost.

The computatioual analysis of the data only depends on the speed of the microprocessor(s) used. When multiple processors are used, their location can have a significant effect on system topology and networking requirements.

Acknowledgment

The authors wish to acknowledge the support given by Transpower New Zealand Ltd in the early phases of this research activity.

References

[1] V. Kuhlmann, "Synchronons measurements and power quality monitoring," Ph.D. dissertation, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Submitted 2004.

[2] T. Skeie, S. Johannessen, and Øyvind Holmeide, "Highly accurate time synchronization over switched ethernet," in *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2001, pp. 195–204.

[3] V. Kuhlmann, A. M. Sinton, M. B. Dewe, and C. P. Arnold, "Effects of sampling rate and ADC width on the accuraey of magnitude and phase measurements in power quality monitoring," 2005, submitted IEEE TPWRD-00173-2005.

[4] IEC 61850-5: *Communication Networks and Systems in Substations. Part 5: Communication Requirements for Functions and Device Models.* International Electrotechnical Commission, July 2003, standard. Ed. 1.0 – English. [Online]. Available: http://www.iec.ch/

Volker Kuhlmann undertook his undergraduate studies at the University of Siegen, Germany, aud gained his M.E. at the University of Canterbury, New Zealand. He is currently completing his Ph.D.

Chris Arnold is a Senior Lecturer in Electrical and Computer Eugineering at the University of Canterbury, New Zealaud. He is a Chartered Engineer and a Member of the IEE. His uain interest is in power system dynamics.

Mike Dewe was Senior Lecturer at the University of Witwatersrand in Johannesburg, South Africa, before joining the University of Canterbury in 1978. He was appointed IT Director in 2000, after having been Associate Professor in Electrical and Electronic Engiueering. He also has a total of 12 years industry experience.

A.8 Implementation Considerations Paper

Submitted to the IEEE Transactions on Power Delivery [112].

Implementation Considerations in Power Quality Instrumentation Design

V. Kuhlmann, M. B. Dewe, C. P. Arnold
University of Canterbury, Christchurch, New Zealand
volkerkuhlmann@gmx.de

Abstract—**Data acquisition systems for power quality and harmonic power flow monitoring have in the past required expensive custom-designed hardware and software. The same quantity and quality of data can now be gathered at much lower system costs.**

This paper presents a number of different data acquisition system configurations and considers their applications, and discusses the need to transfer accurate time information with the data.

Index Terms—**Power quality, power system monitoring, power system harmonics, measurement, monitoring, local area networks, time measurement, synchronization**

I. Introduction

INSTRUMENTS for power quality monitoring are of interest to both electricity suppliers and consumers. Electricity supply networks can be permanently monitored for harmonic power flows, which take up distribution capacity for no real gain. Pollution limits, both legal and agreed, can be observed and policed, and pollution sources can be traced. The instrument's data analysis capabilities should be as versatile as possible, to suit different needs. Ideally, a number of selectable analysis programs which can be commonly expected should be predefined, but there should also be scope for user-defined methods. For fixed installations with predefined scope, instrument versatility can be reduced. Improvements in microcomputer technology and a reduction in size open up new possibilities for instrument design.

This paper examines how technological advances can be used to design more compact instrumentation arrangements at a lower cost. The biggest cost savings can be achieved by implementing functions in software which previously were only possible to be implemented in hardware.

The main requirements for power quality monitoring are continuous measurement and analysis of voltages and currents in power transmission and distribution systems. Of particular interest are harmonic components. Measurement of harmonic magnitudes presents no particular difficulties for the hardware. The recovery of phase angles for harmonic power flow calculations with an accuracy of $1° - 10°$ is more demanding, especially for the higher-order harmonics.

It is essential that sampling is synchronised on all channels. The allowable time error can be as little as $0.1 - 1\,\mu s$. The same synchronisation is required between channels of instruments at geographically different locations.

In order to optimise data storage and retain relevant data, real-time online analysis is required. Ideally, the ability to preset trigger conditions for storing this data is provided.

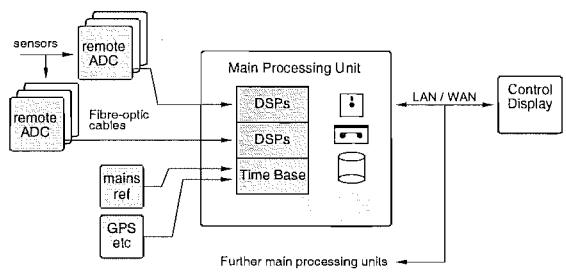


Fig. 1. A typical configuration of a previously developed instrumentation system, which has a hardware time base.

Isolation requirements, as needed in a high-voltage switchyard environment, can also put further constraints on overall system design.

A data acquisition system broadly consists of three stages: A/D conversion, processing/analysis, and storage. The requirements of the ADC stage and the details of modern CPUs and software with respect to power quality monitoring are discussed elsewhere [1], [2].

II. System Architecture

Power quality monitoring places a number of additional constraints on conventional instrumentation systems. Switch yards, for example, are an electrically noisy environment and can produce potentially large voltage differentials in instrumentation systems. Signal sources are likely to be located many metres from a control room. This makes it impractical to carry signals to ADCs located in control/substation enclosures because of the induced noise. Instead, ADCs should be located close to the signal sources. If signal data is carried digitally to the control room, it is relatively immune to noise and further signal degradation is prevented. The best noise immunity on switch yards is obtained with optical communications, which are available at low cost in the form of 100M or Gigabit Ethernet.

The requirements that have by far the greatest effect on the design of the overall system are the need to place ADCs close to the signal sources, and time stamping accuracy [2]. Previously, the cost of the time stamping hardware (GPS receiver, sample clock generation) and of the signal processing hardware caused the concentration of these functions in one base unit. This necessitated the transmission of the sample

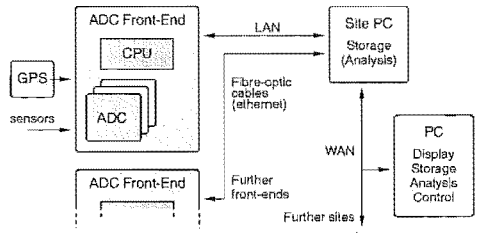


Fig. 2. A typical configuration of a proposed sampling hardware.

clock to the separate ADC units and the return of the data digitally through a noisy environment, as shown again for comparison in the simplified Fig. 1. The remaining part of this section introduces a new design which makes use of technologies which have become available. A block diagram of this new design is shown in Fig. 2. Outlines of further designs emphasising different design goals are also discussed.

For measuring voltages and currents of a 3-phase power distribution system, 6–8 ADC channels are needed. Therefore, to be usable practically, the minimum number of channels in a system is 6. At a sample rate of ≈50 kHz (8× oversampling) and an ADC width of 12 to 16 bit, the resulting 600–800 kbyte/s data rate is well within the capabilities of current CPUs, with computing power to spare.

Oversampling, as in this data rate estimate, increases the bandwidth requirements. More bandwidth is also needed for an anti-alias filter. The advantages of oversampling and filtering are a reduction of computational requirements further on in the processing chain, and a reduction in analog front-end filter component size.

A standard PCI (Peripheral Component Interconnect) bus is clocked with 33 MHz. A 32 bit word can be transferred in one bus cycle in burst mode, which carries an overhead of a few cycles. A single bus access takes approximately 4 bus cycles. This throughput is sufficient for many more than 6 ADC channels. Other researchers are able to transfer in excess of 10 Mbyte/s from custom-made ADC cards with DMA to a hard disk with FFT computations on a P3-800 CPU in a CompactPCI system [3].

Versions of the PCI bus with increased clock speed and 64 bit width have been specified, but have not experienced widespread use in desktop PCs. A new combined serial/parallel bus with 2.5 GHz bus clock, PCI-Express, has a significantly improved bandwidth, and is set to eventually replace PCI. This amount of bandwidth is not likely to be needed by power quality instrumentation.

The more processing that can be performed in the front-end, the less bandwidth and computing is needed at the central processing and analysis stage. However, the particular application will set a limit on how much can be left to the front-end. The tasks of generating the sampling clock and time stamping the samples [2] will need to be performed in any case.

Designing inputs with auto-range hardware has the advantage of effectively increasing the ADC width and allowing a direct connection of a larger range of sensors or other hardware like current or voltage transformers. The disadvantage is that the cost of providing the hardware on the ADC card is significant. The front-end has to provide computing resources for implementing a peak detection of all samples. The input signal is disrupted during a change of range. Whether or not this is an issue depends on the particular application. Programming the system so that the range-selection is fixed is not very different to using a separate input amplifier. The main disadvantage of requiring a programmable input range is that it may effectively preclude using off-the-shelf ADC cards.

If a commercially available ADC card fulfils every other requirement, the need for auto-ranging, and the consequent cost impact, should be re-evaluated. In some circumstances, auto-ranging is not effective, for example where sudden large peaks are to be captured accurately and without being clipped. Auto-ranging is not needed for measuring voltages, or when the approximate current level is known in advance. However, for continuous monitoring of varying current levels, auto-ranging significantly extends the effective ADC resolution. External pre-scalers may be appropriate for the application. These can be put into small cases and be battery-powered, with a mechanical switch operating on three to four signal lines at a time.

Pre-processing the digitised data before transmission to a centralised analysis unit has several advantages. Depending on circumstances, a suitable noise-reduction or low-pass filter can be implemented. Compensation for certain non-linearities of the sensor will make post-processing of data easier. This can be sped up by using simple pre-computed look-up tables. A 16 bit ADC and 32 bit floating point numbers translate into a 256 kbyte table size, which is small by today's standards. While it is not essential to compensate at the ADC front-end, doing so presents filtered and corrected data as output, which reduces complexity and processing at later stages.

At the minimum, it is desirable that the ADC front-end provides at least slightly noise-filtered data which is time-stamped with sufficient accuracy and corrected for sensor non-linearities.

A. *Front-End Design*

Instead of making custom hardware which would likely be uneconomic, front-end CPU systems can be based on one of the commercially available single-board computers. Besides a CPU and memory, these boards often provide at least one PCI slot and various peripheral interfaces, including RS-232C, Ethernet, or IDE hard disk. Intended for use in embedded systems, the form factor is comparatively small, and a low-power version of the CPU is used. CPU throughput is therefore likely to be a little lower than in top-end desktop PCs.

Using, in essence, a PC for this ADC front-end provides further benefits. Single-board computers are commercially available at comparatively low cost. Most of the software development can be done on a PC. Software development systems are much more advanced for PCs and standard tools can be utilised. Development for embedded systems is intrinsically more difficult and requires expensive specialised tools.

If the overall system specifications in terms of time stamping accuracy, bandwidth, or special features like auto-scaling

are such that they can not be met by readily available components, some custom hardware will have to be made. It is desirable that all required custom hardware is integrated on one or more PCI cards, and that if more than one card is needed, they are all identical in design.

For measurements involving more than one geographical location, accepted practise is to use a commercially available satellite time signal receiver as time base, e.g. GPS, GLONASS, or the planned Galileo. These devices supply coarse timing information (1 s and up) via a serial RS-232C interface, and a digital one-pulse-per-second (1 pps) output accurately ($<1\,\mu$s) designating the start of a second, which must be interfaced to the time-stamping circuitry. If the time-stamping is implemented in hardware, the 1 pps signal will have to be supplied to all PCI cards via interconnecting cables. The propagation delay over these cables must either be identical at all locations or small enough to be insignificant.

Flash memory is now available in capacities which are high enough to store the operating system for the front-end CPU, all the application software, and correction tables for a large number of sensors. The system needs to boot from this flash memory. Some single-board systems provide an IDE hard disk interface and/or a compact flash card interface, which any operating system should be able to access easily.

Power supply issues need to be resolved. Batteries can be used for continuous operation up to a certain time limit. For permanent operation, a suitable isolated power supply must be made available.

Environmental issues in terms of temperature, moisture, and electrical noise have to be considered. Unless the equipment is only expected to be used strictly indoors, it needs suitably rugged and moisture proof enclosures. Sealed plastic cases are easily available. Shielding must be sufficient for the electrical noise present in switch yards or substations for reliable operations of digital systems. Appropriate attention needs to be paid to the shielding and noise-immunity of 16 bit ADC circuits or the effective resolution will be reduced. Although an off-the-shelf x86-based computer is by far the most cost-effective computer system available, whether it performs satisfactorily under these circumstances needs to be tested.

When designing equipment, the number of available slots for a bus system is one of the factors that must be considered when deciding about its deployment. CompactPCI offers 8 slots, with additional sets of 8 with the use of bridging circuitry. 3 slots would be taken up by the CPU, GPS, and optical Ethernet cards; the remaining ones are available for ADC cards. Assuming 3 ADC channels per card, a total of 15 channels per front-end can be implemented. This may provide more channels than are practically useful. Any fewer than 3 ADC channels per card is likely not to be cost-effective. This is a further advantage of CompactPCI over standard PC boards, where the number of PCI slots tends to be limited to 5.

B. System Configuration Issues

If more than one front-end ADC system is needed at the same location, the same satellite time source can be used to provide synchronisation. This becomes more economical with a larger number of channels per front-end. It is not as straightforward as having only one time source in the base station. However, the need for very time-critical transfer of ADC clock signals from the base station to the front-end disappears. With continually decreasing cost of GPS receivers, the possibility of providing each front-end with its own GPS receiver becomes a viable alternative.

Data needs to be transferred from the front-end to the base station by some kind of network. The obvious choice is the Internet Protocol (IP) over Ethernet. A 100M Ethernet link provides about 10 Mbyte/s throughput, which is more than sufficient for power quality applications. PCI cards with a fibre-optic 100M (or gigabit) Ethernet interface are readily available, and IP is ubiquitous. USB (universal serial bus) and IEEE1394 are not suitable for local area applications because of their maximum cable lengths of 5 m or less.

III. Networking and Time Transfer

The maximum theoretical bandwidth of 100M Ethernet is $\approx$10 Mbyte/s. If 6 channels produce a total data rate of about 600 kbyte/s, the raw data of several dozen channels can be transferred to a PC over the same Ethernet segment. This would decrease the numerical processing requirements of the front-ends, at the expense of increasing it for a relatively cheap desktop PC. Considering the low cost of dual-CPU PCs, this is certainly a viable alternative.

It can be desirable to use a sampling frequency which is a multiple of the mains fundamental [2]. This necessitates that the actual mains frequency is known by the front-end's sample clock generator, but the measurement of the mains frequency does not have to be performed by the frout-end itself. To reduce the hardware required by the overall system, measurement of the mains frequency can be performed by the site-wide PC to which all the front-ends are connected. The frequency can then be transmitted via the Ethernet link. It is not necessary to do this every mains cycle because the mains frequency change per cycle is small. If measurements are taken over a large system, the frequency at different locations may vary. If this is a problem, measuring the mains frequency on the site PC does not offer any advantage.

A. Time transfers

Instead of providing each front-end with a GPS receiver, a single receiver can be connected to the PC common to a measurement site, as shown in Fig. 3. A time-transfer must then be implemented from the PC to the front-ends. An obvious candidate for this is the network time protocol NTP [4]–[8], which was designed to perform this task over a network with unknown and changeable delays. It compensates for the timing characteristics of the network as much as possible. With the configuration of Fig. 3, the time server would be on the PC, with a client on each of the front-ends.

The signal propagation delays are not negligible, but can be ignored if all sites use the same cable lengths with all front-ends. However, building the system and software with the ability for delay compensation is essentially necessary anyway: when relating data from different sites, the samples

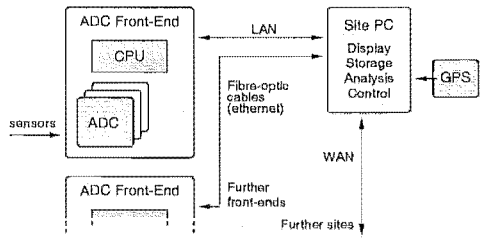


Fig. 3. Another possible configuration of the proposed sampling hardware, utilising the LAN for time transfer. A 1 µs accuracy with this can only be achieved with exceptional effort.

will have to be matched by their time stamps with some closest-fit method. In practice, the delays of packets travelling over the Ethernet link exceeds the signal propagation time by one to three orders of magnitude.

In principle, delays over Ethernet come from the time it takes for packets to enter the transmit queue, wait in the queue, be transmitted, and sit in the receive buffer. The time a packet takes to travel over the wire is proportional to its size, and collisions on the physical link can cause delays. The timing of the interrupts for handling IP packets and the network card also have a large effect on overall network latency.

The characteristic of the Ethernet frame handling largely determines the accuracy of the time transfer. Quality of service features of IP can be used to mark the NTP packets with a higher priority. This reduces latency by moving them to the head of the transmit queue, and therefore increases the accuracy of the time transfer. Any intermediate network devices however would also have to forward these packets at the highest priority. Simple switching hubs (switches) are not capable of doing this, requiring either a system topology not using switches or the use of more expensive switches.

The network diagnostic tool "ping" can be used to obtain an indication of packet transfer speed. A packet is sent and returned by the remote host. The round trip time is displayed, and is ≈250 µs between 2 directly linked x86-500MHz computers, and ≈150 µs for two AMD64-1.8GHz computers under full CPU load linked via a switch. Soliciting a response from the local computer itself cuts the physical link and network card drivers out of the loop; the times are ≈130 µs and ≈40 µs respectively. This shows that a faster CPU reduces Ethernet latency.

To assess the behaviour of NTP over typical networks, a number of NTP installations in active use were queried for their own assessment of accuracy and the networks' timing characteristics. The software in use was SuSE Linux 8.2 with xntp 4.1.1 for both server and client.

Although the NTP jitter is sometimes reported to be as low as 15 µs, this only happens when there is no other network traffic. With moderate network traffic, the jitter is typically reported to be 0.5 – 1 ms. Connecting two computers directly instead of connecting them to a LAN does not improve jitter, which shows that the jitter is mainly caused by the computers' packet handling and task scheduling characteristics.

Over long distances, an accuracy of NTP in the order of 1 – 10 ms seems to be generally accepted. On LANs, reliable synchronisation to less than 1 ms has been demonstrated [8].

Considerable improvement is possible by optimising the Ethernet transport software, and linking it more closely with the NTP software. The basic aim is to prioritise the communication between the NTP client and server. On a multi-tasking system, process-priority for the NTP software can be increased. Network packets associated with NTP can be given higher priority, by tweaking the packet handling and/or by using quality of service features. The largest gain can be achieved by moving the time-critical part of the NTP algorithm, the time stamping of requests, to a high-priority interrupt and possibly even away from the direct control of the operating system. This should not be difficult with a real-time OS or a general-purpose OS with a real-time basis. On a general-purpose OS and PC-like hardware, the time-stamping modifications need to be applied to the low-level Ethernet code.

An accuracy of ±25 µs has been achieved by using hardware time-stamping on the NTP server, and an Ethernet hardware driver on the highest-priority interrupt running independently of the real-time OS [9]. The hardware on the client side was off-the-shelf Ethernet. The NTP server used was integrated into a commercially available network switch which implements time-stamping of packets in hardware and is equipped with a low-drift oscillator. To reach 1 µs accuracy, time-stamping of packets on the client must be performed in hardware. No other network devices are permitted between client and server in either case, and the network is running at 100 Mbit/s (Fast Ethernet) and using standard SNTP protocol [5].

Depending on the level of accuracy desired [10], software and a high-priority interrupt are sufficient. The highest accuracy can only be achieved with specialised hardware on both the client and server, but 1 µs is a possibility.

IV. System Configurations

In the previous sections, advances in processor and network technology were discussed, and possible structures of a new data acquisition system for power quality monitoring outlined. In this section, four different system configurations are outlined with varying trade-offs between performance and cost.

A. Single PC With Standard A/D Card

By far the easiest and cheapest solution is to use a standard PC, commercial ADC cards, and a GPS receiver. The software supplied with the ADC card may not be directly applicable, and further programming for time stamping and for interfacing to GPS may be needed. Time stamping can be performed while the data is being read from the ADC card(s). An accuracy of a few milliseconds should be easily achieved even with a general-purpose operating system. Using a real-time OS would improve the accuracy somewhat, but software development for such a system carries considerable cost. A hybrid real-time/general OS would be a worth-while compromise and should achieve a better than 1 µs accuracy while maintaining the benefits of a general OS. Mains-synchronous sampling is

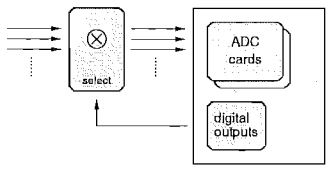


Fig. 4. Using an external programmable pre-scaler and digital outputs to implement auto-ranging.

possible as long as the ADC cards allow fine-tuning of the sampling frequency.

A setup like this would suit deployment indoors, for example in a sub-station building or control room, where the signal sources are close enough to the PC to avoid signal distortion. A UPS (uninterruptible power supply) can be connected if desired. Connection to a LAN is straightforward. The possibility of WAN communications depends on the infrastructure of the particular building.

B. Multiple Enclosures with Off-the-Shelf Hardware

A multiple enclosure system as given in Fig. 2 can also be built using industry-standard hardware. The site PC can have a general-purpose operating system. A large hard disk for storing collected data and an option for archiving/backup is an advantage. An Ethernet interface is now always included. For connecting the front-ends, optical Ethernet cards can be fitted, or a switch with optical connectors can be used instead.

For the front-end computer, a number of different bus systems are available. These should be evaluated in conjunction with the cards commercially available for them. Suitable contenders include CompactPCI, PCI, and PC/104.

The ideal ADC card has 3–6 simultaneously sampled inputs, and a finely programmable sampling frequency. A sampling-clock PLL which can synchronise with an external signal can be used to achieve mains-synchronised sampling, although it would be simpler to program the ADC card with a sampling frequency derived from a mains frequency which is measured elsewhere. The sampling clock is connected to a high-priority interrupt on the CPU for time-stamping. ADC cards of a type for which the relation between a particular sampling clock pulse and the corresponding sample datum can not be established (e.g. delta-sigma converters, or those with FIFOs) are not suitable. Whether auto-ranging inputs are possible depends on the ADC cards featuring programmable input scalers. Some cards include these scalers, but the maximum input voltage is tens of volts at most.

Often a number of simple digital output lines can be found, e.g. from a parallel port or on a CPU or peripherals card. These can be connected to a separate PCB which contains a programmable pre-scaler, as shown in Fig. 4. Any desirable analog filters can be placed on this board too. A place to mount this circuit board can easily be found. It removes one requirement from the ADC card and thus increases the probability of finding a suitable one commercially.

Several small computer systems are on the market, both x86 and PowerPC based, that already include peripherals and flash memory, and provide PCI slots which can be used for interfacing ADC cards. Their power consumption is low to very low.

A media converter between copper and optical Ethernet can be used to remove the need for a PCI slot to hold a fibre-optic card.

C. Custom-Built A/D Converter Card

If suitable ADC cards are not commercially available, or not at an acceptable price, a custom-designed card should be considered. Apart from the ADC converters themselves, a number of other parts are needed as well. The bus interface is best implemented with a commercially available chip. Several models are readily available for PCI and CompactPCI. Availability of drivers for this chip for the operating system to be used is of high importance to reduce development time and therefore cost.

A variable frequency oscillator for the sampling clock is not necessarily an important feature. For reasons of coherency between channels, it is better to have only one clock source shared by all ADC cards and all channels. This clock can be generated on one of the ADC cards and distributed to the others, but generating the clock by the CPU might be more appropriate. It is important that the ADCs can be switched to an external clock source (which may come from the CPU). Mains-synchronised sampling can be achieved by programming an appropriate clock frequency. It is paramount that the circuitry is designed in such a way that the relationship between individual clock pulses of the sample clock, and the sample data that is read from the card afterwards, can be correctly established. Furthermore, it must also be possible to relate each sample clock pulse to the system's method of time keeping with an accuracy of better than 1 µs. The higher silicon integration available today makes it possible to place all digital circuitry needed for an ADC card into a single FPGA (field programmable gate array).

If analog filters and input pre-scalers are deemed necessary, they should be placed on the ADC card if possible. It is necessary to find an appropriate balance between the maximum number of channels which fit on the card, the maximum card area available, and the higher CPU-time that is required to achieve the same amount of filtering, if filtering is performed by the CPU instead of on the ADC.

A very strong argument in favour of PCI is that the cards can also be used in standard PCs. This would be an excellent and cost-effective approach for any software development, especially if the operating systems used on the front-end and the development PC are more or less identical. This would be the case for Linux, Linux with a real-time base, and MS-Windows with a real-time base, but not for dedicated real-time operating systems from various embedded systems software companies.

Being able to use the ADC cards in standard PCs also opens up the possibility of a number of different system configurations. When it is not necessary to have the front-ends separate from the site PC as shown in Fig. 2, they can be combined into one unit and deployed in a similar manner to

that discussed in section IV-A. The issues for interfacing time signal receivers are identical to those for separate front-ends and those for a system as in section IV-A.

Implementing the ADC card in CompactPCI instead of PCI form factor opens up more possibilities for deployment in front-end systems as shown in Fig. 2 or Fig. 3. The disadvantage is that using them in a standard PC requires a suitable adapter.

D. Custom-Built Hardware

Designing every piece of hardware in-house gives the greatest flexibility for the design of the time stamping and sampling circuitries, and the largest possible freedom for selection of each individual component. All the circuitry fits into a contemporary FPGA.

A mostly custom-build front-end system offers the highest integration density, although it is not necessarily much higher than that of a CompactPCI system. Designing a GPS receiver in-house would not be cost-effective. Receiver modules with a serial interface and 1 pps output packed on a small circuit board should be bought in. It is also unlikely to be economic to make CPU cards or modules. Some already have memory and a bus interface integrated.

Designing the ADC card component of the system, as outlined in section IV-C, should be considered for the possibility of deployment in different situations. The ADC design issues discussed in section IV-C also apply for a full custom design.

Custom-designing and manufacturing any hardware is expensive in both development time and development equipment such as logic analysers or in-circuit emulators. It is only justifiable either if a system which meets the specifications can not be assembled from commercially available parts, or if a high number of systems is to be produced. For high production numbers, ASICs (application specific ICs) are cheaper than FPGAs.

V. Conclusion

It was shown that a modern power quality monitoring system can now be built much more simply and cheaply. In particular:

- Custom-designed components can largely be replaced with industry-standard ones.
- Small volume systems can be designed, greatly increasing portability. For example, the electronics for 6–12 ADC channels can be fitted into a CompactPCI box approximately $100 \times 160 \times 80\,\mathrm{mm}^3$, excluding power supply, battery, and shielding.
- Especially for applications which require time stamping of the order of 10–100 µs rather than 1 µs or less, it is possible to generate the time signal at a single processor and transfer it to each distributed front-end over an Ethernet LAN.
- GPS unit price has been greatly reduced, opening up the possibility of using multiple receivers instead of distributing precise time (1 µs) over a LAN.

As described, a number of different system configurations are possible, providing the required functionality for a range of applications. Such low-cost data acquisition systems will allow a greater use of informative quality monitoring of power systems.

Acknowledgment

The authors wish to acknowledge the support given by Transpower New Zealand Ltd in the early phases of this research activity.

References

[1] V. Kuhlmann, A. M. Sinton, M. B. Dewe, and C. P. Arnold, "Effects of sampling rate and ADC width on the accuracy of magnitude and phase measurements in power quality monitoring," 2005, submitted, TPWRD-00173-2005.

[2] V. Kuhlmann, C. P. Arnold, and M. B. Dewe, "Relevance of microprocessor advances for power quality instrumentation," Sept. 2005, in preparation.

[3] M. P. Hayes, "A real-time beam steering system for KiwiSAS-IV," in *Proceedings of Electronics New Zealand Conference (ENZCON2003)*, Hamilton, New Zealand, Sept. 2003, pp. 27–32. [Online]. Available: http://www.elec.canterbury.ac.nz/research/acoustics/papers/2003/Hayes_enzcon_2003a.pdf

[4] D. L. Mills, "Network time protocol (version 3) specification, implementation," RFC 1305, Mar. 1992. [Online]. Available: ftp://ftp.ietf.org/rfc/rfc1305.pdf

[5] D. Mills, "Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI," RFC 2030, Oct. 1996. [Online]. Available: ftp://ftp.ietf.org/rfc/rfc2030.txt

[6] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, 1991.

[7] ——, "Improved algorithms for synchronizing computer network clocks," *IEEE/ACM Trans. Networking*, vol. 3, no. 3, pp. 245–254, 1995.

[8] ——, "Adaptive hybrid clock discipline algorithm for the network time protocol," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 505–514, 1998.

[9] T. Skeie, S. Johannessen, and Øyvind Holmeide, "Highly accurate time synchronization over switched ethernet," in *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2001, pp. 195–204.

[10] IEC 61850-5: *Communication Networks and Systems in Substations. Part 5: Communication Requirements for Functions and Device Models.* International Electrotechnical Commission, July 2003, standard. Ed. 1.0 -- English. [Online]. Available: http://www.iec.ch/

Volker Kuhlmann undertook his undergraduate studies at the University of Siegen, Germany, and gained his M.E. at the University of Canterbury, New Zealand. He is currently completing his Ph.D.

Mike Dewe was Senior Lecturer at the University of Witwatersrand in Johannesburg, South Africa, before joining the University of Canterbury in 1978. He was appointed IT Director in 2000, after having been Associate Professor in Electrical and Electronic Engineering. He also has a total of 12 years industry experience.

Chris Arnold is a Senior Lecturer in Electrical and Computer Engineering at the University of Canterbury, New Zealand. He is a Chartered Engineer and a Member of the IEE. His main interest is in power system dynamics.

A.9 Microprocessor Advances Paper

Submitted to the IEEE Transactions on Power Delivery [108].

Relevance of Microprocessor Advances for Power Quality Instrumentation

V. Kuhlmann, C. P. Arnold, M. B. Dewe
University of Canterbury, Christchurch, New Zealand
volkerkuhlmann@gmx.de

Abstract—**In the past, data acquisition systems for power quality monitoring required expensive custom-designed hardware and software. With recent advances in standard PC components, this is no longer the case.**

The effectiveness of standard CPUs and operating systems in fulfilling power systems instrumentation functions is evaluated. A CPU register and interrupts can be used for sample clock generation and time stamping. Modern CPUs include signal processing instructions that used to be limited to specialised DSPs, and general-purpose operating systems such as Linux provide most, if not all, of the necessary software features.

Index Terms—**Power quality, power system monitoring, power system harmonics, synchronization, signal sampling, data processing, microprocessors, microcomputers, signal processing, sampled data systems**

I. Introduction

ADVANCES in microprocessor performance bring with them new opportunities for instrumentation systems in terms of size, cost, and bandwidth.

Previously, time-stamping samples with an accuracy of 1 µs required the use of dedicated hardware. This could only be achieved with custom-built hardware, which is time-consuming to develop and expensive to build.

Where one digital signal processor (DSP) was required per input channel some years ago, a single processor can now handle multiple channels. The biggest savings can be achieved when it becomes possible to handle the sample clock generation and time stamping by software, possibly with the help of timing registers in the processor or the peripherals.

The main requirements for power quality monitoring are continuous measurement and analysis of voltage and current vectors in power distribution systems. Of particular interest are harmonic components. A sampling rate of 5 kHz is sufficient for the 50th harmonic magnitude (50 Hz system), which does not place a high demand on hardware. The recovery of phase angles for harmonic power flow calculations with an accuracy of $1° - 10°$ is more demanding, especially for the higher-order harmonics. An even higher demand is placed on equipment when transients are to be examined.

It is essential that the sampling of input values occurs simultaneously on all inputs, so the resulting values, which may for example represent several 3-phase systems in a switch yard or across a power network, can be correlated with each other. The upper limit of allowable time error between inputs is determined by the application, and is typically in the order of 1 µs.

First, the effect of the substantial CPU speed increases in recent years is discussed. This is followed by a series of code execution time measurements on a number of different processors. Section IV considers factors about time stamping accuracy, sampling rate and synchronisation, that need to be considered for power quality monitoring. Software aspects and the suitability of different types of operating systems are discussed, and a general-purpose OS is shown to be an effective option. Issues for system design, including ADC, front-end and possible system configurations are dealt with elsewhere [1].

II. Processor Architectures

Despite the 3 orders of magnitude improvement in microprocessor performance in the last 20 years, interrupt latency has only improved by one order of magnitude. Fortunately this improvement is sufficient to satisfy all but the most stringent power quality timing requirements.

A downside from the point of view of precise timing is the indeterminism of instruction execution times as a side effect of overall performance improvement. Execution times become mostly statistical averages.

A commonly-used technique for embedded systems is to lock a code sequence into the internal cache. Some processors, e.g. PowerPC, allow this. Although it reduces the overall available cache size, it guarantees that the code which permanently resides in the cache is executed at maximum speed, and at a known rate.

If the processor is used for time-critical tasks, any non-deterministic execution of CPU instructions becomes irrelevant if the amount of indeterminism is less than the timing accuracy required. This now appears to be the case with modern processors.

The remaining part of this section introduces the main characteristics of a selection of microprocessors which can be used to build instrumentation systems.

A. *Intel Pentium/Xeon and* AMD *Athlon/Opteron*

Originally designed as a 16 bit CPU approximately 25 years ago, this family has proven very popular. Multi-processor capable versions of the current 32 bit and 64 bit models are available from both Intel and AMD.

A significant disadvantage of this family of processors is the requirement for backwards compatibility. This has resulted in a complex design with a large number of transistors,

TABLE I
EXECUTION TIMES OF A 128 POINT COMPLEX FFT ON VARIOUS PROCESSORS, USING THE FFTW LIBRARY AND ITS "BENCH" PROGRAM.

CPU	single	double	optimisations
G4 1.25 GHz	2.0 µs	4.4 µs	fma, altivec
P3 450 MHz	13.9 µs	14.1–14.4 µs	sse
P4 mobile 1.8 GHz	1.93 µs	2.89 µs	sse / sse2
P4 2.4 GHz	2.83 µs	3.31 µs	—
	1.52 µs	2.13 µs	sse / sse2
Athlon 1800+ MP (1.53 GHz)	2.29 µs	3.30 µs	—
	2.34 µs	3.33 µs	sse / sse2
	1.27 µs	3.31 µs	k7 (3dnow)
Opteron 244 MP (1.8 GHz)	1.83 µs	2.75 µs	sse / sse2
	1.07 µs	2.76 µs	k7 (3dnow), sse / sse2
Xeon 2.4 GHz MP	2.94 µs	3.34 µs	—
	1.40 µs	2.17 µs	sse / sse2
ultrasparc 296 MHz	14.06 µs	13.86 µs	

TABLE II
EXECUTION TIMES OF COMPLEX FFTS WITH VARYING NUMBER OF POINTS, USING FFTW. THE HARDWARE IS AS FOR TABLE I.

CPU	points	single	double	optimisations
P3:	128	8.02 µs	14.3 µs	sse / —
	256	16.6 µs	32.1 µs	sse / —
	512	34.8 µs	74.6 µs	sse / —
	1024	92.7 µs	205 µs	sse / —
Athlon:	128	1.27 µs	3.34 µs	k7 / sse2
	256	3.84 µs	7.15 µs	k7 / sse2
	512	8.04 µs	16.25 µs	k7 / sse2
	1024	17.3 µs	35.3 µs	k7 / sse2
Opteron:	128	1.07 µs	2.75 µs	k7 / sse2
	256	3.16 µs	5.86 µs	k7 / sse2
	512	6.65 µs	13.5 µs	k7 / sse2
	1024	14.4 µs	29.0 µs	k7 / sse2

and hence high power consumption. A key reason for the family's continued development is binary compatibility with legacy software. For any new designs not requiring this legacy compatibility, alternative families are often chosen.

However, despite these disadvantages the x86 family represents excellent value for money due to high-volume production. Software support for these processors is also excellent due to their ubiquitous nature. A standard PC can prove to be cost-effective for almost all phases of product development.

B. Power PC

The PowerPC architecture, by IBM and Motorola, is a RISC (reduced instruction set computer) design with simple and modern features, and a low power consumption. The register layout is straightforward: there are 32 integer and 32 floating point registers, all with identical functionality.

The PowerPC is commonly used for embedded systems in appliances which require reasonably high computing power, such as laser printers. It provides multiply/add instructions which speed up signal processing computations. Such instructions were previously only found in DSPs. 64 bit versions of the CPU are now available, and multi-processor systems are supported. Versions for embedded systems typically have several peripherals integrated on the CPU chip, for example network interface, memory and bus controllers. Effectively this is a computer-on-a-chip, and is ideal for new designs of flexible and extensible instrumentation equipment.

The PowerPC is well-supported by development systems and operating systems, including Linux. Virtually every real-time OS also has a version for PowerPC.

C. Other General Purpose CPUs

The Acorn Risc Machine (ARM) is a very low power CPU family with a reasonable computing speed. It is worth considering for new designs, but it is only used in embedded systems and not very commonly.

A large number of micro-controller families are available for use in embedded devices, but they lack the computing power needed for multi-channel continuous harmonic analysis.

D. DSPs

Digital signal processors are primarily intended for certain numerical computations on comparatively high volumes of data. Their instruction sets have always been optimised to allow efficient implementation of numerical algorithms.

However, the DSP has simply become superfluous for this type of application. A single general-purpose processor provides sufficient processing capabilities for a medium-sized power quality monitoring system. There are first-class compilers available for at least the x86 and PowerPC families (and one of the best is free of charge). Choosing a general-purpose CPU is more flexible and offers a much wider range of available off-the-shelf hardware and software.

III. BENCHMARKS

Performing FFTs is one of the major computational tasks for a power quality monitoring system. Benchmarks for a number of common processors were performed using the Fftw library [2] and its benchmarking program. The library is available for several architectures. A special characteristic of Fftw is that it performs a number of tests at runtime on the target system to construct code which runs fastest under actual runtime conditions. All tests were performed with 128 data points and with complex numbers, for both single and double sized floats (floating point numbers). A "single" float is 4 bytes long (32 bits), and a "double" float 8 bytes (64 bits). 128 points were chosen because they are sufficient for measurements up to the 50th harmonic.

The Apple powerbook with the G4 processor was operated under its native Mac OS X. The Sparc system was running Solaris 7. All other machines were running Linux with a kernel around version 2.4.20 and with the C compiler gcc 3.2 or 3.3. Threading was disabled on those computers with two CPUs.

The results are shown in Table I. The benchmark program was run several hundred times for each combination of CPU and options, and the results combined into a single figure. As with any benchmark, results depend on many parameters and should always be treated with due caution.

When applied to a power supply system, a new FFT has to be calculated every 20 ms (50 Hz system). Modern processors

are more than three magnitudes faster than that. All contemporary processors complete the task in under 5 µs. An outdated P3-450 is still under 15 µs. These execution times need to be multiplied by the number of channels the instrumentation system is able to handle.

Even if the number of points is increased to 1024, computation time remains within acceptable limits. As Table II shows, a 1024 point complex FFT on a P3-450 takes about 200 µs at double precision. Neither double precision nor a 1024 point transform length are needed for power quality applications.

Therefore, the question of which processor is best for use in a power quality monitoring system should not be answered by processor speed at all, because they all provide adequate performance. Questions of cost and the availability of development systems should be the most important factor, although the cost of the development system proportionally reduces with larger production volumes. The P3 should not be ruled out, because it also provides adequate performance, and might be available on boards for embedded systems with a low-power version. For a desktop system, however, the speed differences are significant.

There should be ample scope for handling 48 channels, a modern operating system with data I/O functions for storage and network communications, and the generation of sampling clocks and time stamps on just one processor. A 64 bit CPU is unnecessary, and a 32 bit Athlon, P4, or PowerPC is sufficient. The best cost/performance ratio is offered by the Athlon, but ultimately other factors like power consumption or availability of off-the-shelf processor cards need to be considered when designing instrumentation systems [1].

IV. Sample Timing Issues

Synchronous measurements require that the sampling on each input occurs at precisely controlled instants. The maximum allowable uncertainty for this depends on the particular measurement being undertaken, and is often much shorter than the time between samples.

For distributed power quality monitoring or harmonic state estimation, the time stamping accuracy of the sampling typically needs to be in the order of 0.1–1 µs. For example, 1° of phase angle at the 50th harmonic is approximately equivalent to 1 µs. Therefore, to achieve this, the samples need to be time-stamped with an accuracy better than 1 µs. For fault location using wave-front timing methods, an accuracy of 0.2 µs is desirable [3].

Synchronisation of sampling, or channel coherency, is achieved by clocking all ADCs (analog-to-digital converter) with the same signal. The ADCs must have a known and constant time after which the result appears at the outputs. The following tasks need to be performed:

- time stamping
- sample clock generation
- sampling at a multiple of the fundamental frequency (to reduce filter and FFT computational requirements by eliminating windowing issues).

It should be possible to perform all these 3 tasks by utilising a 32 bit counter register provided by the CPU. Each time the counter reaches its maximum value, a sample pulse is generated. The sampling frequency can be adjusted by changing the maximum counter value.

Time-stamping is performed by generating another interrupt with the one-second output of a suitable time reference unit (for example, a satellite time receiver).

The mains frequency can be derived from the values of the counter at the time of the mains fundamental zero crossings, if this zero crossing signal is made available to the CPU. Alternatively, the mains frequency can be derived from one of the power lines being measured.

Of critical concern is the timing and latency of the interrupts, because this affects the counter values being read. The time receiver interrupt should be assigned highest priority. The sample clock interrupt is not time critical because the whole sampling period is available for saving the sample data and calculating a time stamp from previously saved counter values.

An 80 MHz Power PC CPU has an interrupt latency in the order of 1–2 µs. 400 MHz versions are available, with correspondingly lower latencies. A large part of the latency is deterministic and can therefore be compensated for. An accuracy of < 1 µs can be achieved without great difficulty.

It is therefore possible to perform sample clock generation and time stamping in software with the help of a suitable CPU register, while only placing a small to moderate load on the CPU, leaving capacity for processing of the data.

V. Software Considerations

A. Operating Systems

For centralised data processing the choice of software is reasonably straightforward. The sample data is already time-stamped by the ADC front-end when it arrives for processing. None of the tasks are time-critical, and communications via Ethernet provide in themselves a certain amount of buffering. Desktop CPUs provide ample power for number crunching, with slight differences between manufacturers (see section III). Multi-CPU systems, or symmetric multi-processor (SMP) systems, are available and cost-effective. A general-purpose operating system is well suited. Linux has proven to be reliable and would be a good choice for a dedicated instrumentation system. In particular, it enables users to log in remotely, which allows full use and control of the instrumentation system and its software without having to be on site.

Using a general-purpose operating system (OS) enables the sharing of data analysis and visualisation software between the central processing station and any other PC which carries a copy of the sample data or pre-processed sample data. It also makes data archival on various media trivial.

For the ADC front-end system, which has to perform the sample clock generation and time stamping, any of the real-time operating systems for embedded systems could be used. Although these systems guarantee a response within a certain time, their average throughput can be rather low, and when resources (other than CPU time) are depleted, they simply stop working altogether.

In contrast, general purpose operating systems continue when resources are depleted, but at a much lower speed.

In the case of memory exhaustion, this is made possible by configuring virtual memory (swap space). Under these conditions the instrument can no longer perform the intended task adequately, but the OS still allows the user to log in and fix the problem. This is an important consideration for an autonomous system which can be operated remotely.

For a data acquisition system, the real-time requirements are rather simple. With an external signal, counter values or ADC output values need to be saved within a short time frame. With a 5 kHz sampling rate (50th harmonic at 50 Hz), ADC data of each channel must be saved every 400 µs. This interval is too short for regular process scheduling and therefore this task has to be performed in an interrupt service routine (ISR). Sampling rates much higher than this necessitate the use of DMA (direct memory access) or a CPU with sufficiently low interrupt latency. Even with higher sampling rates, it is not impossible to run a general-purpose OS.

A general-purpose OS can be run on a real-time kernel [4], but this is more sophisticated than is required. A sufficient solution is to modify the Linux kernel and to provide a fast interrupt routine; the availability of the source code makes this possible. Versions of Linux for embedded systems already exist, but the basics of a regular Linux system easily fit into currently available flash memory modules and RAM sizes.

Software for storing the programs, and exchanging control information, programs and data with the base system, is already taken care of by a general-purpose OS, simplifying that part of the design considerably.

The fastest execution of the ISR which handles the time stamping can be achieved by permanently reserving one or two CPU registers for use by this ISR only. Reserving the registers saves the time of saving their contents on entry to the ISR and restoring them on exit from the ISR. They can be treated as read-only by the remaining code. Short of writing all the code in assembler (which is impossible for a project of this scope) the compiler has to be told not to generate code which makes use of the reserved registers. With the GNU C compiler gcc [5] it would be possible to modify the target CPU's machine description and reduce the number of available registers, and then to recompile the compiler. A condition for this is that the CPU has a reasonably large number of identical registers, which is true, for example, for the PowerPC. As the remaining code has to run in fewer registers, a slight performance penalty has to be expected.

For a non-centralised data processing configuration, processing is performed by the ADC front-end itself, which requires more CPU power and leaves the central system to deal with storage, display and archiving.

B. Development Environments

A significant aspect for consideration is the choice of development tool for the system, especially for the front-end. Software for the centralised processing on a "site PC" can be developed with whichever development tools are usually used for software running on the operating system used for the site PC.

There are advantages in using the same OS on the front-end as on the site PC, if possible. If so, much of the software can be developed and tested on any PC, especially if the ADC cards used at the front-end can also be plugged into a PC.

Numerous operating systems are on the market for embedded systems, either real-time, or general purpose with and without real-time extensions. The latter category includes Linux and MS-Windows. These are typically sold with an integrated development system, and the development systems often run on a range of host systems and offer a range of target systems. Some companies require royalty payments for each copy of their embedded software in use whereas others do not.

C. Programming Languages and Application Software

Scripting languages promise rapid prototyping at a high level. They can also be used to extend applications to users with an easy means to interface short user-programs [6], for example for data analysis, to the control application. An example of using Python for message passing (i.e. data exchange) exists [7]. Many of these issues apply to the scenario of controlling multiple power quality measurement systems from a common station.

Kale [8] compares various scripting and other programming languages. Scripting languages have a reputation for introducing large computing overheads and being slow. While this is almost certainly true for some, Python seems to show that this need not be the case [9].

D. Analysis Software

A general-purpose instrumentation system should be as flexible as possible as far as the analysis software is concerned.

A user-selectable set of analysis functions can be provided as building blocks, or plug-ins. Plug-ins for common applications should be built into the system, and flexibility can be provided by allowing for easy addition of user-designed plug-ins. Both built-in and user-designed plug-ins are implemented with a defined function API (application program interface). On a general-purpose OS, shared libraries are commonly used, and can be loaded by the main program any time after startup. Other techniques are also suitable.

Plug-ins can include the functions:

- reading sample data
- FFT
- transient detection
- saving critical values which exceeded a threshold
- generating alerts when pre-set conditions are met

Each plug-in has an associated "cost" of CPU and memory bandwidth use. The total cost of all building blocks combined by the user must stay below 100%, with 100% being the system performance left over after the requirements for the time stamping and the operating systems are deducted from the total provided by the hardware.

Further possible analysis functions, which can also be implemented, include:

- capturing and analysing harmonics that exceed certain boundaries, or are outside certain shapes. Also capturing to either side to show buildup and the dynamic reaction of the grid.

- calculating power quality phenomena including flicker according to published standards [10]–[12], and logging and generating an alert when agreed or preset thresholds are exceeded.
- recording details when line voltages are outside their nominal range.
- recording details of any transients, including voltages and currents for a short period before and after the event.
- keeping a continuous log of line frequency and transferred power.
- keeping a permanent record of those grid parameters which were pre-selected by the user.

If this instrumentation system is to be connected to the internet, security considerations need to be taken very seriously. This becomes paramount if the front-ends are connected via the site PC and the internet to a central PC. The encryption which will need to be used for this incurs a non-negligible CPU load penalty, which must be considered and the site PC dimensioned accordingly. Experience has shown repeatedly that internet security issues have not received appropriate attention.

VI. Conclusion

A single modern CPU can now be used for continuous synchronous measurements of voltage and current vectors in power distribution systems, instead of expensive hardware-based solutions. In particular:

- Improvements in CPU performance mean that DSPs are no longer required. The improvements in performance have been associated with a lesser improvement in latency, but latency is now low enough for most, if not all, power quality measurements.
- FFT benchmarks indicated that a 32-bit Athlon, P4 or PowerPC processor is adequate to handle up to 48 data channels, plus data storage, network communications, sampling clock generation and time stamping.
- A 32-bit counter register within the CPU can be used for time stamping and to control the sampling frequency.
- A real-time operating system is not required, and general-purpose operating systems are simpler to work with. General-purpose operating systems based on a real-time core are a good compromise for applications which benefit from a limited amount of real-time functionality.

Acknowledgment

The authors wish to acknowledge the support given by Transpower New Zealand Ltd in the early phases of this research activity.

References

[1] V. Kuhlmann, M. B. Dewe, and C. P. Arnold, "Implementation considerations in power quality instrumentation design," Sept. 2005, submitted, TPWRD-00581-2005.

[2] FFTW (Fastest Fourier Transform in the West) is a C subroutine library for compnting the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data. [Online]. Available: http://www.fftw.org/

[3] M. B. Dewe, S. Sankar, and J. Arrillaga, "The application of satellite time references to HVDC fault location," *IEEE Trans. Power Delivery*, vol. 8, no. 3, pp. 1295–1302, 1993.

[4] J. Odagiri, N. Yamamoto, and T. Katoh, "Porting EPICS to L4-Linux based system," in *Proceedings of the 2001 Particle Accelerator Conference (PAC 2001)*, vol. 2, 2001, pp. 1183–1185.

[5] "Gcc, the GNU compiler collection." [Online]. Available: http://gcc.gnu.org/

[6] F. Basegmez, "Extending a scientific application with scripting capabilities," *IEEE Comput. Sci. Eng.*, vol. 4, no. 6, pp. 52–59, 2002, [see also IEEE Compntational Science and Engineering].

[7] D. M. Beazley and P. S. Lomdahl, "Extensible message passing application development and debugging with Python," in *Proceedings of the 11th International Parallel Processing Symposium*, 1997, pp. 650–655.

[8] L. V. Kale, "Programming languages for CSE: The state of the art," *IEEE Comput. Sci. Eng. Mag.*, vol. 5, no. 2, pp. 18–26, 1998, [see also Computing in Science and Engineering].

[9] L. Prechelt, "An empirical comparison of seven programming languages," *IEEE Computer*, vol. 33, no. 10, pp. 23–29, 2000.

[10] *IEEE Recommended Practice for Monitoring Electric Power Quality*. New York, USA: The Institute of Electrical and Electronics Engineers, Inc., 1995, IEEE Standard 1159-1995.

[11] IEC 61000-4-30: *Electromagnetic Compatibility (EMC). Part 4-30: Testing and Measurement Techniques—Power Quality Measurement Methods*. International Electrotechnical Commission, Feb. 2003, standard. [Online]. Available: http://www.iec.ch/

[12] IEC 61000-3-7: *Electromagnetic Compatibility (EMC). Part 3: Limits. Section 7: Assessment of Emission Limits for Fluctuating Loads in MV and HV Power Systems—Basic EMC Publication*. International Electrotechnical Commission, Nov. 1996, standard. [Online]. Available: http://www.iec.ch/

Volker Kuhlmann undertook his undergraduate studies at the University of Siegen, Germany, and gained his M.E. at the University of Canterbury, New Zealand. He is currently completing his Ph.D.

Chris Arnold is a Senior Lecturer in Electrical and Computer Engineering at the University of Canterbury, New Zealand. He is a Chartered Engineer and a Member of the IEE. His main interest is in power system dynamics.

Mike Dewe was Senior Lecturer at the University of Witwatersrand in Johannesburg, South Africa, before joining the University of Canterbury in 1978. He was appointed IT Director in 2000, after having been Associate Professor in Electrical and Electronic Engineering. He also has a total of 12 years industry experience.

appendix B

Chartdat Manual and Reference[1]

By Volker Kuhlmann, 24 June 1997.
This document describes chartdat version 1.4, 21 June 1997.

B.1 Introduction

This manual describes the chartdat tool for analysing sample data files generated by the CHART data acquisition system. It complements the existing postcess tool. Postcess can display data graphically as well as output it in ASCII. Chartdat does not generate graphs, but it has various functions which can be applied to the data. As well as outputting data and filtered data in ASCII form, chartdat can also display all information stored about the data, which is useful as documentation of the measurements undertaken. New functionality can be added.

Chartdat is a command-line oriented tool, which makes it easy to port to other platforms, including those with a faster processor than an Intel 80x86. Ports to Linux (for platforms 80x86, Alpha, and Sparc), Solaris on Sparc, and MS-DOS are available. Ports to other platforms would be easy to develop but have not been done due to lack of equipment.

Because of the absence of a graphical user interface (GUI) the chartdat tool remains compact and powerful without the limitations a GUI often imposes on applications. It makes it also comparatively easy to program platform-independently.

This manual makes no attempt to explain all the values found in CHART-generated data files but instead focuses on the operation of chartdat.

Although chartdat is strictly speaking not a filter program, it can best be thought of as being such. A filter program reads data (input), processes it, and then writes the processed data back to disk (output). CHART data is always stored in two separate files, one containing the actual sample data and the other defining various operating parameters. It is impossible to read the CHART data from the standard input channel (stdin), or write CHART data on the standard output channel (stdout), because these channels are equivalent to only one file each. Therefore, when reading or writing CHART data, chartdat needs to operate on actual files, not re-directable I/O-channels, which is against the definition of a filter program. As shown later, chartdat still makes use of the stdin and stdout channels.

[1] This appendix contains the text of reference [107].

B.1.1 Invoking Chartdat

On practically all operating systems in use today, programs are run from a command prompt by simply typing their name. Parameters are typed after the program name, and are separated by one or more spaces from the name and from each other.

Example:

`chartdat -help`

Runs program `chartdat` with the argument `-help`. (Which will cause chartdat to display its help.)

Parameters containing spaces must usually be quoted to avoid the one parameter being interpreted as two. Parameters are also called options or arguments. When they are called options it usually means that this parameter is optional.

B.1.2 Help

When chartdat is run without arguments, a very brief synopsis of the purpose of chartdat is displayed, and that the integrated help text is available with the `-help` parameter.

Like most parameters, the `-help` parameter has a long version and a short version. The short version, `-h`, is faster to type, but `-help` is easier to remember. Both work in exactly the same way.

The help screen of chartdat is reproduced in figure B.1. It shows all parameters with their long and short names, the syntax of the additional number, number range, string, or time needed by some parameters, and some limitations for this particular version of chartdat. The help output is a brief summary of sections B.2, B.3, B.4, B.5, and B.6.

B.2 Principal Operation of Chartdat

The principal operation of chartdat is to apply one main function to the CHART data, and optionally one or more restrictions to the applied function. The applied main function is called an *action*, and the available restrictions are called *limits*. Actions are explained in section B.4, and limits in section B.5. These sections describe the available actions and limits in detail. How these functions can be usefully combined is shown in section B.7.

Chartdat is, as a command line program, controlled by parameters given to it on the command line. Some of these parameters are interpreted by chartdat as commands, some as arguments to these commands, and some as filenames. The usage (section B.1.2) gives a complete overview, and detailed information is in sections B.4, B.5, and B.6. All frequently used instructions to chartdat have a short and a long form. There is no functional difference between them. They are shown as in `-h|-help`, which means that either `-h` or `-help` can be used. The `|` should never be typed.

The order of the parameters to chartdat is unimportant except that filenames must be given last. Commands to chartdat which require an argument for themselves, like the "threshold" limit which needs the threshold value, must have this argument immediately following separated by one or more spaces. The "copy" action needs a second filename (output filename), this can either be specified with the `-o|-out` option (section B.6) as in

```
chartdat ... -out outfilename infilename
```

or given behind the first file name (no `-o` or `-out` is needed) as in

```
chartdat ... infilename outfilename
```

.

All normal output of chartdat is written to the standard output channel (stdout). This can be redirected into a file with the > operator on most operating systems, as in

```
chartdat ... > filename
```

. Any error messages are written to the standard error channel (stderr). The "limittimes" limit

```
General purpose data manipulation program for use with data generated by the
CHART (Continuous Harmonic Analysis in Real-Time) system.
University of Canterbury, EEE Dept, Christchurch, New Zealand.
Volker Kuhlmann
VK  V1.4  21 June 1997

Usage:
       chartdat ACTION LIMITS OPTIONS FILE
       chartdat LIMITS OPTIONS -c|-copy -o|-out OUTFILE INFILE
       chartdat LIMITS OPTIONS -c|-copy INFILE OUTFILE

Actions: (must give exactly one of these)
  -i|-info              display info about a def and dat file
  -c|-copy              copy a def and dat file set
  -v|-view              view the contents of a def and dat file
  -vh|-viewhead         as -view, but only display headers, no data
  -vt|-viewtime         display times only of data packets
  -vn|-viewnum          display data only of data packets, 1 number per line
  -s|-stats             as -vh, also display mean, etc. of each data packet
  -fix                  fix def and dat file (e.g. if not closed/updated)

Limits for actions: (any number of these or none)
  -th|-thresh FLOAT     limit to packets exceeding this threshold value
  -ti|-times TIMERANGE  limit to this time range
  -ls|-limitsets SETS   data sets to which an action is limited
  -lt|-limittimes       limit to this list of times (created by -viewtimes?)
                        (list of times is read from standard input)
  SETS       comma-separated list of numbers or RANGEs
  RANGE      two hyphen-separated numbers (empty numbers default to min/max)
  TIMERANGE  two hyphen-separated times (empty numbers default to min/max)
             format: '123456' (GMT), or '31 01 1999 23:59:59' (local)

Options:
  -h|-help              show help and exit
  -o|-out OUTFILE       specify output filename
  -ve|-verbose          show progress
  -D|-debug             print debugging info
  --                    stop processing arguments - only names remaining

Remaining arguments are taken as filenames.
Never specify the 'def' or 'dat' extension!
Can not read data from stdin (or write to stdout) because 2 files (def, dat)
are required.
Current limits:
   def file headers (data storage sets):   20
   data sets (store data sets):            200

Further information can be found in the documentation.
```

Figure B.1: The help output of chartdat (when called with -h).

reads a list of times from the standard input channel (stdin). To connect stdin to a file and therefore read the list of times from this file, most operating systems use the < operator, as in

```
chartdat ... < filename
```

CHART data is always stored in two files, referred to as DEF/DAT file pair. The file extensions are required to be `.def` and `.dat`. When giving CHART data filenames to chartdat, however, never specify these extensions as chartdat appends them. So

```
chartdat ... chart_measurements
```

would operate on the files `chart_measurements.def` and `chart_measurements.dat`.

There is a limit to the number of store data sets (def file headers) and data sets chartdat can handle before it runs out of memory. The limits are compiled into chartdat, and are displayed together with the help (near the end), see section B.1.2 and figure B.1. Currently, CHART uses 1 store data set and up to 20 data sets. The UNIX version of chartdat can handle up to 20 store data sets and up to 200 data sets. Therefore, this limitation of chartdat should be irrelevant in practice.

B.2.1 Return Codes and Error Messages

Whenever chartdat terminates it returns an exit status to the invoking program, that is the user's shell. It is possible to use this exit code to react to any errors chartdat might have encountered. The exact procedure to do this is operating system dependent.

The exit codes used are tabled here:

Exit code	Label	Description
0	EXIT_OK	No error. Normal termination.
2	EXIT_USAGE	Usage was requested.
3	EXIT_ERRVALUE	A given value was incorrect (either out of range, or wrong format).
4	EXIT_NOVALUE	A value or parameter is missing.
5	EXIT_NOACTION	No action was specified. Exactly one must be given.
6	EXIT_EOF	End of file was encountered. This is not necessarily an error!
7	EXIT_ERRIO	Input or output error.
8	EXIT_CORRUPT	Corrupted data was read.
9	EXIT_NOMEM	Not enough memory available to complete the operation.

Every time chartdat terminates the exit code is printed by chartdat.

Chartdat tries to display a meaningful error message for every error situation. Because the standard C function ferror() does not always give useful messages, an additional message is given by chartdat.

B.3 Time and Time Specification

This document mentions the term *coarse time* in various places. The time stamp associated with each data packet consists of two values: *coarse time* and *vernier time*. The vernier time stores the part of the time with a resolution smaller than 1 s. Coarse time stores the part of the time from 1 s and above.

Within CHART, calendar times and dates are not stored as local time, but as a time in time zone GMT, which has an offset of 0. This makes it possible to compare times universally between time zones, and avoids confusion with daylight saving times. The time (system time) is stored as a *time_t* value, which means it is a count of the number of seconds elapsed since 1 Jan 1970 00:00:00 GMT. Times before then and after 19 Jan 2038 03:14:07 GMT can not be represented with this format, because it is a 32-bit integer value. The coarse time in the data packets is a time_t value.

When converting the time_t value to local time, the local time offset must be added. For a country in time zone GMT+12h, 12 h must be added, assuming no daylight saving is in effect.

Whenever a complete time (second, minute, hour, day, month, year, time zone) is required by chartdat from the user, the format must be:

```
day month year hh:mm:ss
```

The time zone can not be specified and is always the local one for chartdat. It is important that the local time zone is correctly configured. The month can be given as number, 3-letter abbreviation, or full name. Case is insignificant. The year must have the century — 97 is not the same as 1997 (97 will produce a range error). The time is always a 24 h time.

One variation to the time format is allowed: it can be given in the time_t format, as decimal number, hexadecimal number (0x12345678), or octal number (with a leading 0). These time_t values are not very human-readable, but they can nevertheless be useful. There can be little misunderstanding about time zones and daylight saving when using time_t values. They always represent a time in the GMT time zone, never in a local one. Chartdat automatically recognises if the specified time is a time_t value. Whenever chartdat outputs a time with a 32-bit number next to it, it is the time_t representation of the clear text time in the local time zone, as indicated by the time zone specification in the time.

Example:

> Wed 20 Nov 1996 02:13:47 GMT
> Wed 20 Nov 1996 15:13:47 NZDT
> 0x3292695b, 0848456027
>
> All four values specify the same point in time. The first time in in GMT, the second in local time, New Zealand Daylight saving Time in this case. It can be seen that NZDT = GMT + 13 h. The last two numbers are the corresponding time_t value, in hexadecimal and decimal notation.

The program timeconv (by Volker Kuhlmann) aids in converting times between local and GMT time zones and time_t values, taking care of daylight saving.

B.4 Actions

This section describes the available actions of chartdat.

All of these actions can be limited. Some actions might be of limited use if not limited.

info (-i|-info)
Displays all information available about the DEF/DAT file pair file pair specified. This includes all information about the data sets, as well as the number of packets stored in the DAT file belonging to any of these data sets, and a few other minor things like file size. See figures B.2 and B.3 for an example output.

copy (-c|-copy)
Copies the specified DEF/DAT file pair to a new DEF/DAT file pair. It is especially useful if used with one of the limits in section B.5.

Currently the whole DEF file is copied, independent of any limits. This means that the resulting DEF file might contain data sets which are not present in the DAT file.

The new DEF/DAT file pair can be specified with the `-o` or `-out` option, or as the second file name on the command line.

view (-v|-view)
Displays the contents of the specified DEF file first, then the packet headers and packet data of all packets in the specified DAT file. Example output is shown in figure B.4.

view head (-vh|-viewhead)
As for "view", but only the DEF file and the packet headers of the DAT file are shown. No data of the data packets is listed. Output is as in figure B.4 but no data is displayed.

view times (-vt|-viewtimes)
This displays the coarse time value of each packet of the DAT file. One time value per line is generated. No other output is produced.

The format of this output is as required for the "times list" limit and described in section B.3. Example output is shown in figure B.6. The time_t value is shown first, then " = ", then the corresponding local time.

view numbers (`-vn|-viewnum`)
Shows nothing but the data, one number per line. This format is meant to make it easy to import the numbers into a spreadsheet or other graphing software.

statistics (`-s|-stats`)
Similar to "view head". Displays the DEF file, the header of each packet in the DAT file, and some statistics on each data packet. Currently the mean, variance, and standard deviation are computed. Statistical values are computed for each packet individually. Example output is shown in figure B.5.

fix (`-fix`)
Fixes an inconsistent DEF/DAT file pair.

If the active data sets are not deleted before shutting down the PPU, the DAT file is not closed and some values in the DEF file are not updated. Most of the data is still useful, but some packets might be missing at the end of the DAT file, and the last packet in the DAT file will almost always be incomplete.

This action truncates[2] the DAT file after the last complete packet, and recomputes the missing values in the DEF file out of the data, thus avoiding errors due to inconsistent data.

It does not make sense to limit this action, therefore, any limits given will silently be ignored.

No output is produced if the files were consistent. Any changes which were made are displayed, as in figure B.7.

Warning: this action alters the specified DEF/DAT *file pair!*

B.5 Limits

This section describes the available limits of chartdat. Any one, including none, of these limits can be applied to any of the actions explained in section B.4, unless otherwise stated. The description of these limits is meant to be complete.

The general effect of these limits is to restrict the action chosen to certain data packets.

threshold (`-th|-thresh FLOAT`)
Limits the action to those data packets which have at least one value exceeding the given threshold. For FLOAT insert the threshold. It is a floating point number. Usual rules for formats of floating point numbers and computers apply.

time range (`-ti|-times TIMERANGE`)
Limits the action to those packets which fall within the given time range, inclusively (interval [start-stop]). Give the time range in place of TIMERANGE as two hyphen-separated times. The time format of each time is given in section B.3 and can be either a time_t value or clear text time. If the first time is missing it is assumed to be the earliest possible time, if the second time is missing it is assumed to be the latest possible time.

data sets (`-ls|-limitsets SETS`)
Limits the action to those packets belonging to one of the given data sets. Give the data sets in place of SETS.

The data sets are given as their number. The specification can be a single number, a comma-separated list of numbers, a number range, or any combination of number lists and number ranges. A number range comprises two numbers separated by

[2] Some operating systems, including MS-DOS, do not have a file truncate function. In this case the DAT file will be left untruncated.

a hyphen ("-"). If the left number is missing, it is assumed to be the minimum allowed value, if the right number is missing it is assumed to be the maximum allowed value.

times list (-lt|-limittimes)
Limits the action to those packets which have a coarse time equal to one of the times in the times list. The times list is a list of times, one per line, and is read from the standard input channel (stdin). Only time_t values are accepted here, and they must be at the beginning of the line. The remainder of the line is ignored. The format of the time_t specification is given in section B.3.

B.6 Other Command Line Options

This section describes those command line arguments which are neither actions nor limits.

help (-h|-help)
Displays the integrated help text (also shown in figure B.1).

output DEF/DAT file pair (-o|-out FILENAME)
This option can be used with the "copy" action and specifies the base name of the DEF/DAT file pair data is copied to. Do not specify the ".def" or ".dat" extension as they will be added by chartdat. For FILENAME above insert the actual name.

verbose (-ve|-verbose)
This shows some additional progress output during the operation of chartdat. The output generated might be of limited use but the function is supplied anyway.

debug (-D|-debug)
This option is used during the chartdat software development phase. Its use is discouraged and at own risk.

last argument (-)
This option means that it is the last option on the command line, all following strings are file names. This allows to process files which happen to have a name equal to one of the actions, limits, or options.

B.7 Examples

This section gives some examples of the use of the various capabilities of chartdat. It is assumed that the CHART data is stored in files called `chartmeas.def` and `chartmeas.dat`.

Display all information about the data, but no data itself:

```
chartdat -info chartmeas
```

Produce similar output to the example above but show only data set 1:

```
chartdat -info -limitsets 1 chartmeas
```

Other valid specifications of a data set limit are:

```
-limitsets 1,3,5
-limitsets 1-4,9-12
-limitsets 6-
```

Display some statistics on every packet of data sets 1, 3, and 5. Both lines are equivalent:

```
chartdat -stats -limitsets 1,3,5 chartmeas
chartdat -limitsets 1,3,5 -stats chartmeas
```

Search for packets of data sets 2, 4, 6 having at least one sample exceeding the given threshold. Headers of the packets are displayed:

```
chartdat -viewheads -limitsets 2,4,6 -thresh 7.55 chartmeas
```

Copy data sets 1 to 3, 6, and all data sets from 12 and above. Both lines are equivalent:

```
chartdat -copy -limitsets -3,6,12- chartmeas destname
chartdat -copy -limitsets -3,6,12- -out destname chartmeas
```

Copy all packets between 14:00 and 16:00 on 3 March 1997, local time. Both commands are equivalent:

```
chartdat -copy -times "3 3 1997 14:00:00-3 3 1997 16:00:00"
                                                    chartmeas destname
chartdat -copy -times 0x331a2290-0x331a3eb0 chartmeas destname
```

List the times of all packets of data sets 2 and 4 exceeding the threshold:

```
chartdat -viewtimes -limitsets 2,4 -thresh 3.2 chartmeas
```

As above but storing them into a file:

```
chartdat -viewtimes -limitsets 2,4 -thresh 3.2 chartmeas >timesfile
```

List the headers of all packets having a time equal to one of the times in the file timefile:

```
chartdat -viewhead -limittimes chartmeas <timesfile
```

B.8 Future Improvements

As possible improvements to chartdat, these items are suggested:

1. The ability to merge multiple sets of DEF/DAT file pair s into one.
2. Allowing more than the one format in section B.3 for specifying times.
3. Is it useful to display times in local time as well as in GMT? The program timeconv can be used instead.
4. Currently the times list for the "limittimes" limit must be sorted. The packets of the DAT file must be sorted as well (this restriction is currently fulfiled by all CHART-generated DAT files). This restriction could be removed, but the increase in computational requirement must be considered.
5. The "copy" action currently copies the whole DEF file independent of any restrictions. If for example the data set limit is used, the resulting DAT file does not have all data sets contained in the resulting DEF file. Whether this imposes any problems remains to be seen.

```
------------------------------------------------------------------------------
File: /tmp/ripple/tp/rtpi3a.def
      Papanui T.P. ripple magn + phase Currents 66kV
Data sets: 8              (head code  0)
------------------------------------------------------------------------------
   |DataSetRef:   1              (head code  1)
   |Start:        Mon 18 Nov 1996 19:00:10 NZDT  =  0x328ffb6a
   |Stop:         Tue 19 Nov 1996 15:59:41 NZDT  =  0x3291229d
   |Descr.:       Papanui ripple phase 66kV AP100 CB92 ISL CCT2 Red Current
   |Data type:    Ripple
   |X, Y Unit:    , Degree
   |Packets:      7558         Length: 1000          StoreRate:  10.000
   |Min Acq, Upd: (            ), (            )    DispColour: 255 0 0
   ------------------------------------------
   |DataSetRef:   2              (head code  1)
   |Start:        Mon 18 Nov 1996 19:00:10 NZDT  =  0x328ffb6a
   |Stop:         Tue 19 Nov 1996 15:59:41 NZDT  =  0x3291229d
   |Descr.:       Papanui ripple magn 66kV AP100 CB92 ISL CCT2 Red Current
   |Data type:    Ripple
   |X, Y Unit:    , Volt
   |Packets:      7558         Length: 1000          StoreRate:  10.000
   |Min Acq, Upd: (            ), (            )    DispColour: 255 0 0
   ------------------------------------------
   |DataSetRef:   3              (head code  1)
     .    ...
     .
     .
```

Figure B.2: A sample display of the contents of a DEF file. This information is displayed by the "info", "view", "viewhead", and "stats" actions.

```
File: /tmp/ripple/tp/rtpi3a.dat
Size: 33682432, read ok: 33682432, trailing junk: 0
DataSetRef Packets    Description & times
1          4144       Papanui ripple phase 66kV AP100 CB92 ISL CCT2 Red Current
                      Start time: Mon 18 Nov 1996 19:00:10 NZDT  =  0x328ffb6a
                      Stop time:  Tue 19 Nov 1996 06:30:40 NZDT  =  0x32909d40
2          4144       Papanui ripple magn 66kV AP100 CB92 ISL CCT2 Red Current
                      Start time: Mon 18 Nov 1996 19:00:10 NZDT  =  0x328ffb6a
                      Stop time:  Tue 19 Nov 1996 06:30:40 NZDT  =  0x32909d40
3          4144       Papanui ripple phase 66kV AQ100 CB102 ISL CCT3 Red Current
                      Start time: Mon 18 Nov 1996 19:00:09 NZDT  =  0x328ffb69
                      Stop time:  Tue 19 Nov 1996 06:30:40 NZDT  =  0x32909d40
4          4144       Papanui ripple magn 66kV AQ100 CB102 ISL CCT3 Red Current
                      Start time: Mon 18 Nov 1996 19:00:09 NZDT  =  0x328ffb69
     .    ...
     .
     .
```

Figure B.3: Output produced by the "info" action, after the DEF file contents (as in figure B.2) have been displayed.

```
Serial pckt #: 1
DataSetRef:    6
CoarseTime:    0x328ffb69 (GMT)  =  Mon 18 Nov 1996 19:00:09 NZDT
PrecisionTime: 0x00000001           1
FundFreq:      50.02
Description:   Papanui ripple magn 66kV McFadden1 Red Current
DataType:      Ripple
XUnit, YUnit:  , Volt
StoreRate:     10.00
  0.000014  0.000008  0.000007  0.000005  0.000006  0.000012  0.000009  0.000004
  0.000007  0.000005  0.000004  0.000007  0.000007  0.000006  0.000014  0.000005
     .    ...
     .
     .
```

Figure B.4: Output produced by the "view" action. The DEF file information is displayed first, as in figure B.2. The "viewhead" action produces the same output as the "view" action, except that it shows none of the data.

```
Serial pckt #: 1
DataSetRef:    6
CoarseTime:    0x328ffb69 (GMT)  =  Mon 18 Nov 1996 19:00:09 NZDT
PrecisionTime: 0x00000001          1
FundFreq:      50.02
Description:   Papanui ripple magn 66kV McFadden1 Red Current
DataType:      Ripple
XUnit, YUnit:  , Volt
StoreRate:     10.00
N:                      250
Mean:                   0.000008759
Variance, Std deviation: 0.000000000, 0.000004797

Serial pckt #: 2
DataSetRef:    5
     .    ...
     .
     .
```

Figure B.5: Output produced by the "stats" action. The DEF file information is displayed first, as in figure B.2.

```
0x328ffb69  =  Mon 18 Nov 1996 19:00:09 NZDT
0x328ffb69  =  Mon 18 Nov 1996 19:00:09 NZDT
0x328ffb69  =  Mon 18 Nov 1996 19:00:09 NZDT
0x328ffb69  =  Mon 18 Nov 1996 19:00:09 NZDT
0x328ffb6a  =  Mon 18 Nov 1996 19:00:10 NZDT
0x328ffb6a  =  Mon 18 Nov 1996 19:00:10 NZDT
0x328ffb6a  =  Mon 18 Nov 1996 19:00:10 NZDT
0x328ffb6a  =  Mon 18 Nov 1996 19:00:10 NZDT
0x328ffb73  =  Mon 18 Nov 1996 19:00:19 NZDT
0x328ffb73  =  Mon 18 Nov 1996 19:00:19 NZDT
     .    ...
     .
     .
```

Figure B.6: Output produced by the "viewtimes" action. No DEF file or any other information is displayed. Times listed more than once came from data packets with the same time but corresponding to different data sets.

```
DataSetRef 1          : Updating no of packets (7558 -> 4144)
DataSetRef 1          : Updating stop time (3291229d -> 32909d40)
DataSetRef 2          : Updating no of packets (7558 -> 4144)
DataSetRef 2          : Updating stop time (3291229d -> 32909d40)
DataSetRef 3          : Updating no of packets (7558 -> 4144)
     .    ...
     .
     .
```

Figure B.7: Output produced by the "fix" action if fixes were made.

appendix C

The DSM Schematic Diagrams

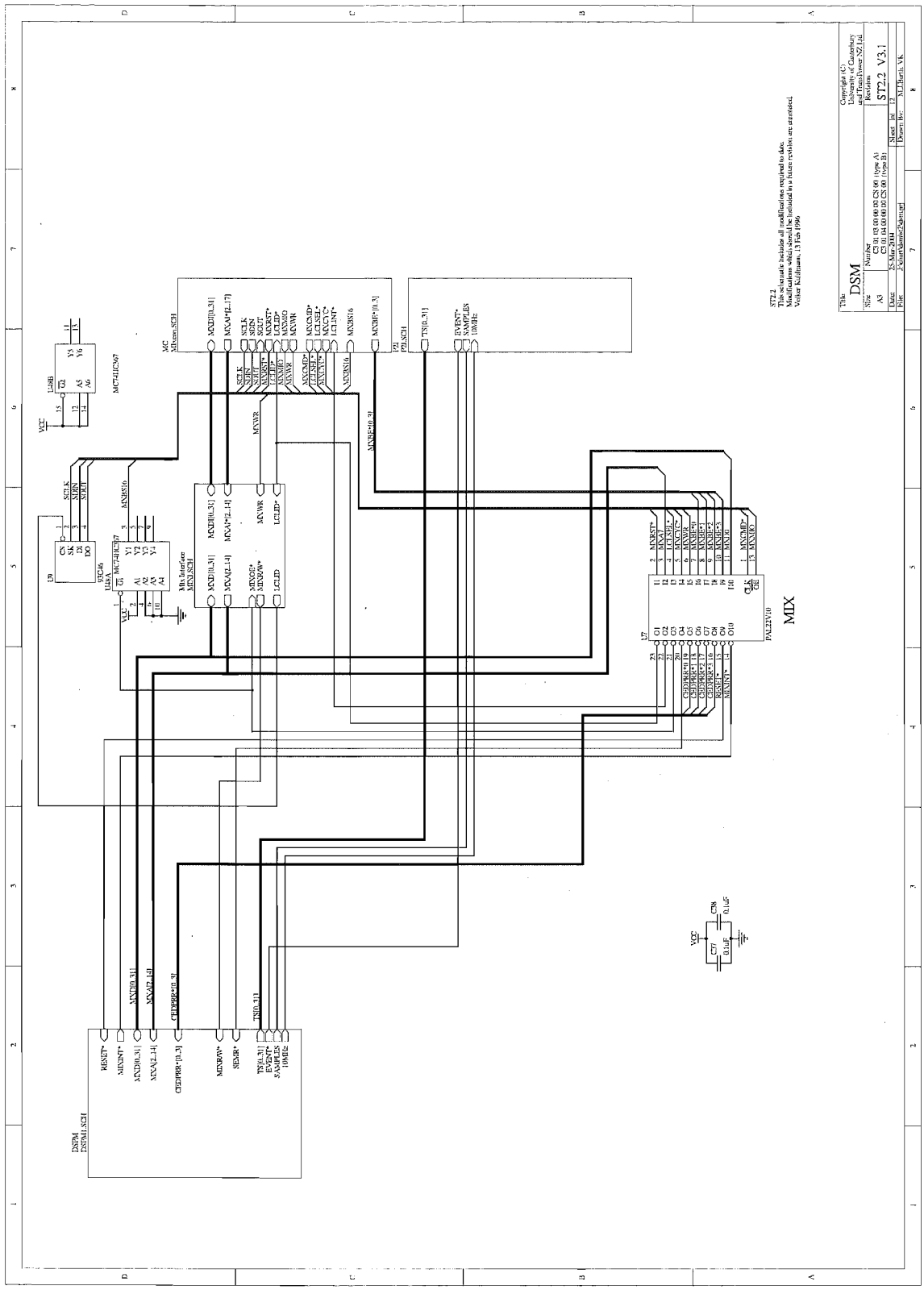
In the following the schematic diagrams for the DSM are shown, using a Texas Instruments digital signal processor.

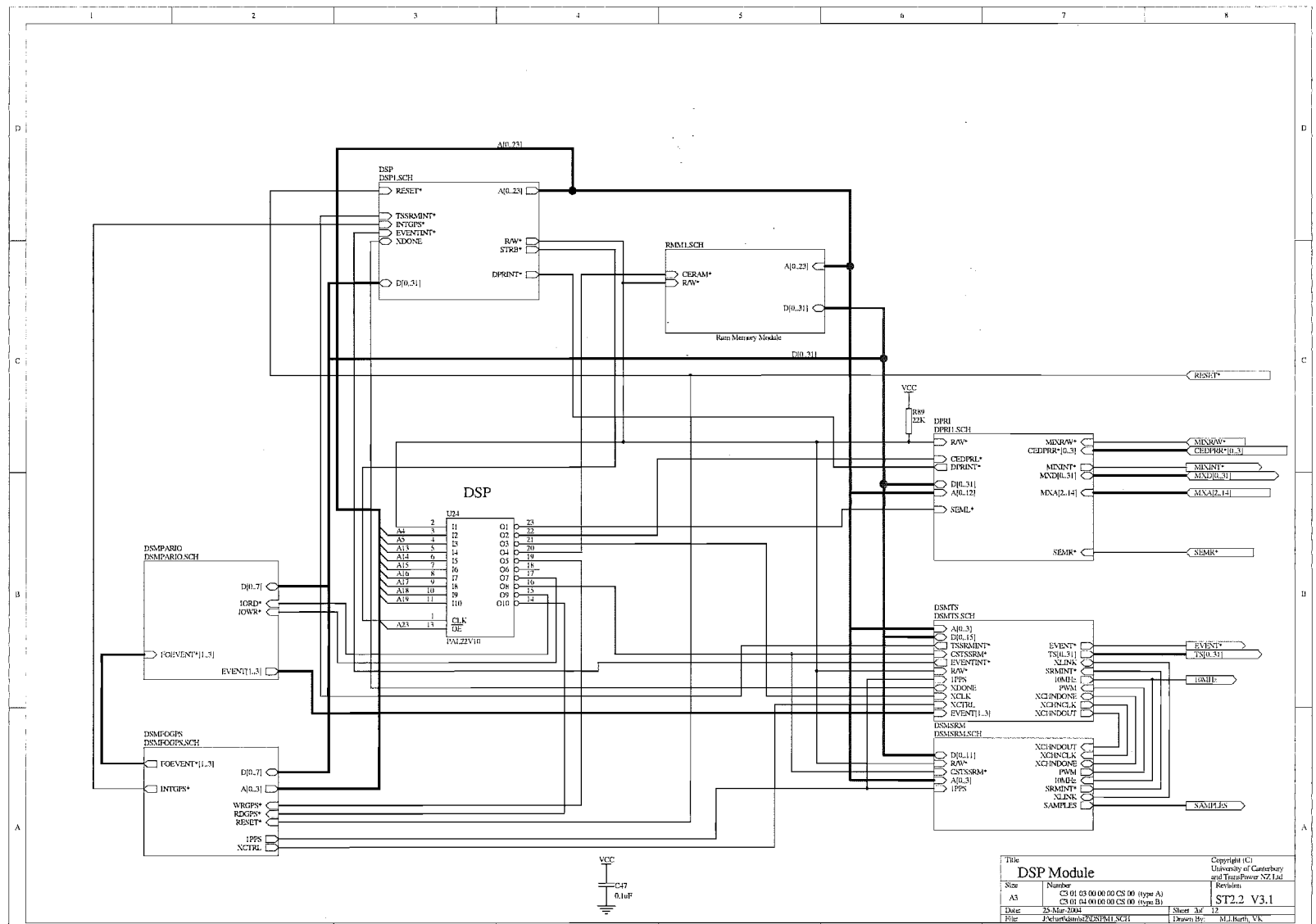
A schematic for a cable connecting a Magnavox MX4200 GPS receiver to the DSM is given on page 216.

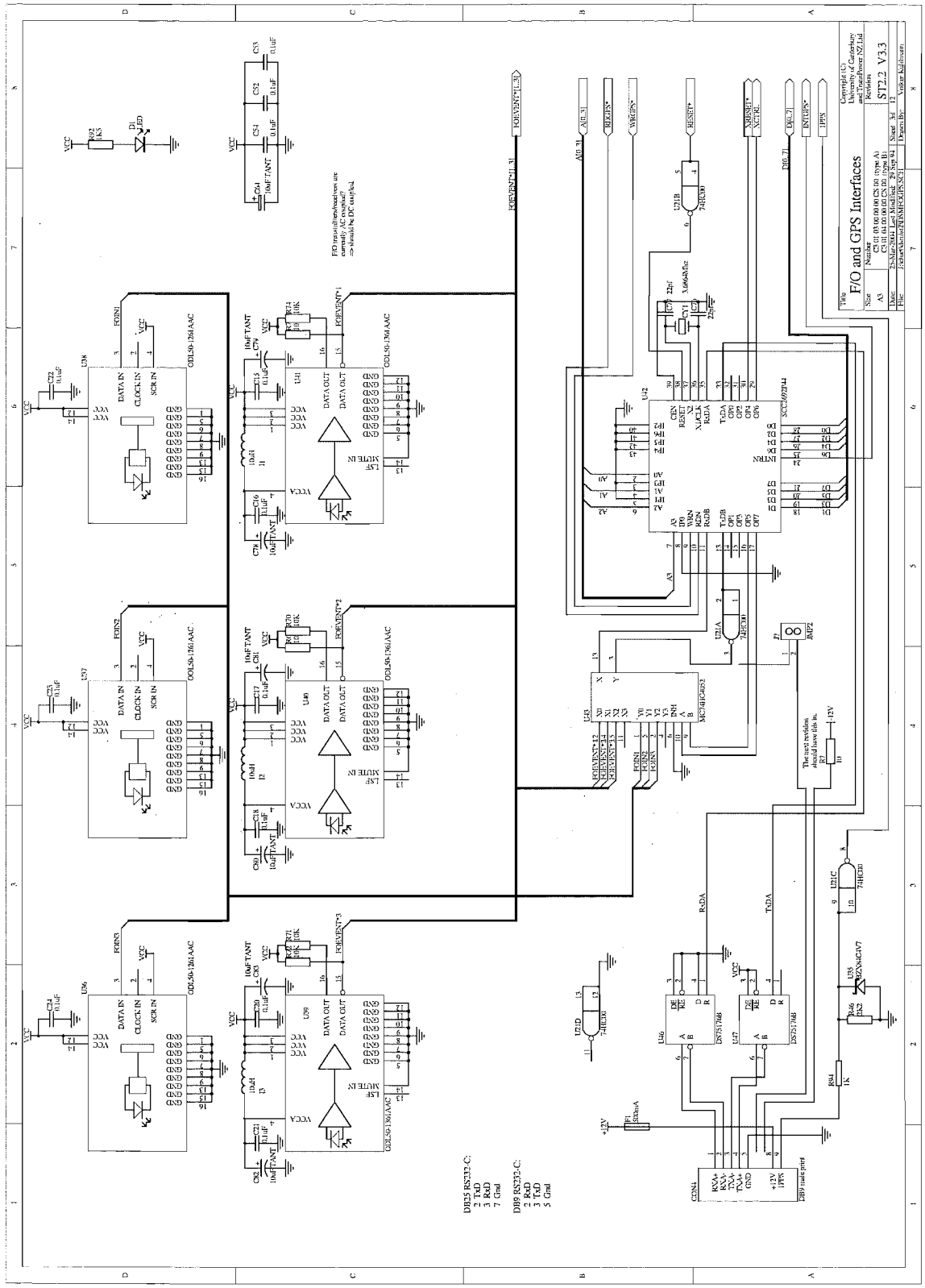
Tables showing the pinout of the Multibus II connectors P1 and P2, and the jumpers, memory map, and FPGA registers of the DSM are given in appendix C.2 starting on page 217.

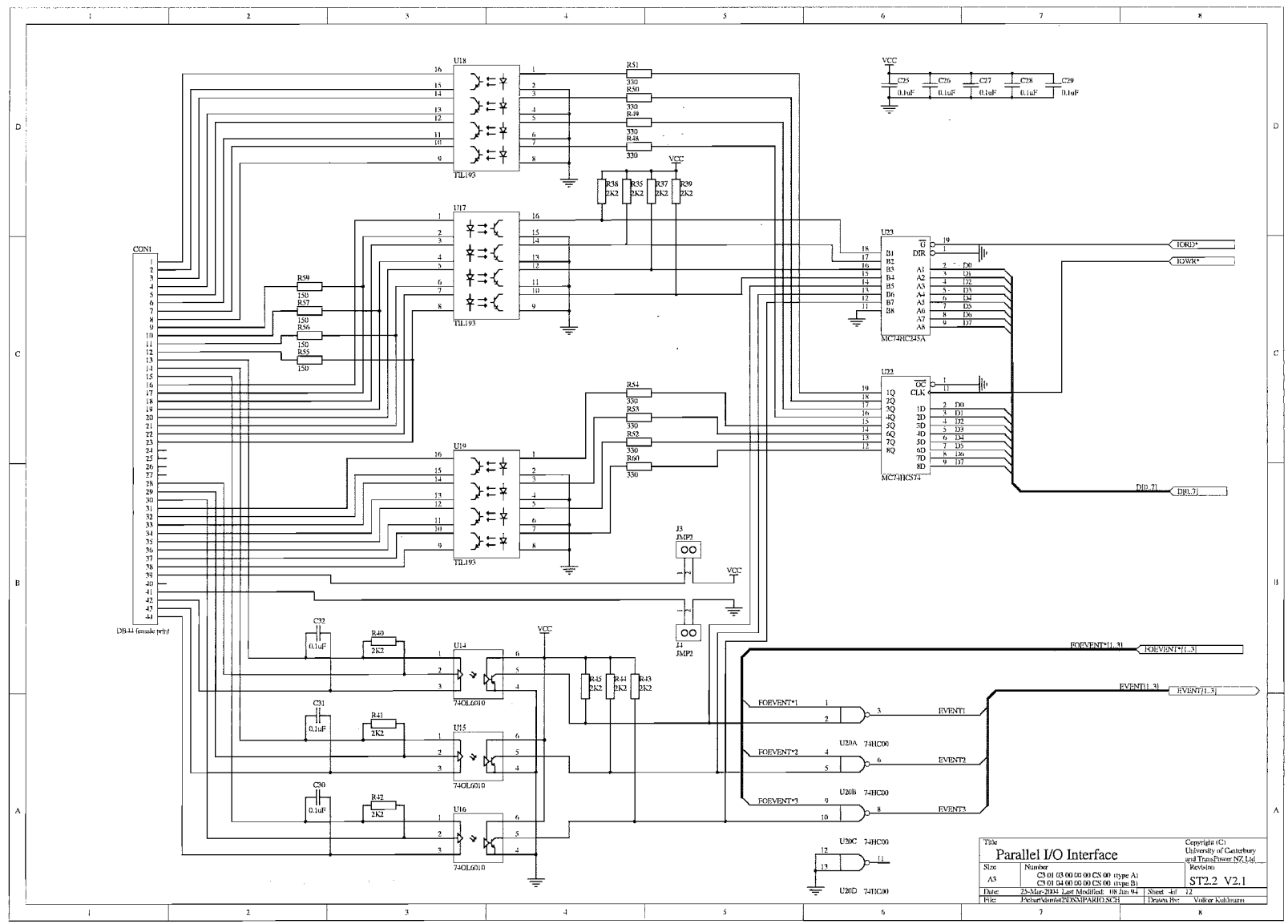
C.1 The DSM with TMS320C31

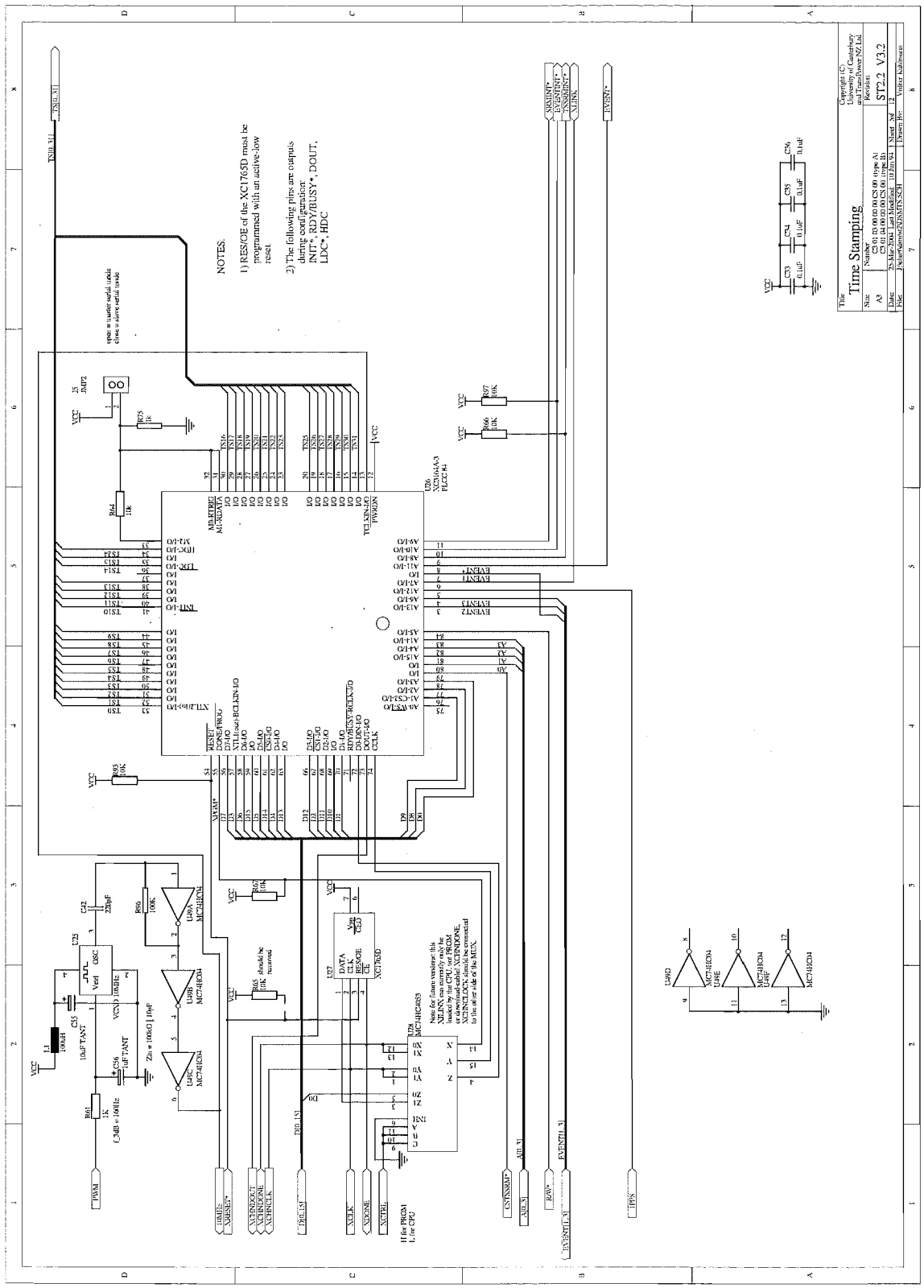
The schematic diagrams of the DSM using a Texas Instruments TMS320C31 DSP are on the following pages.

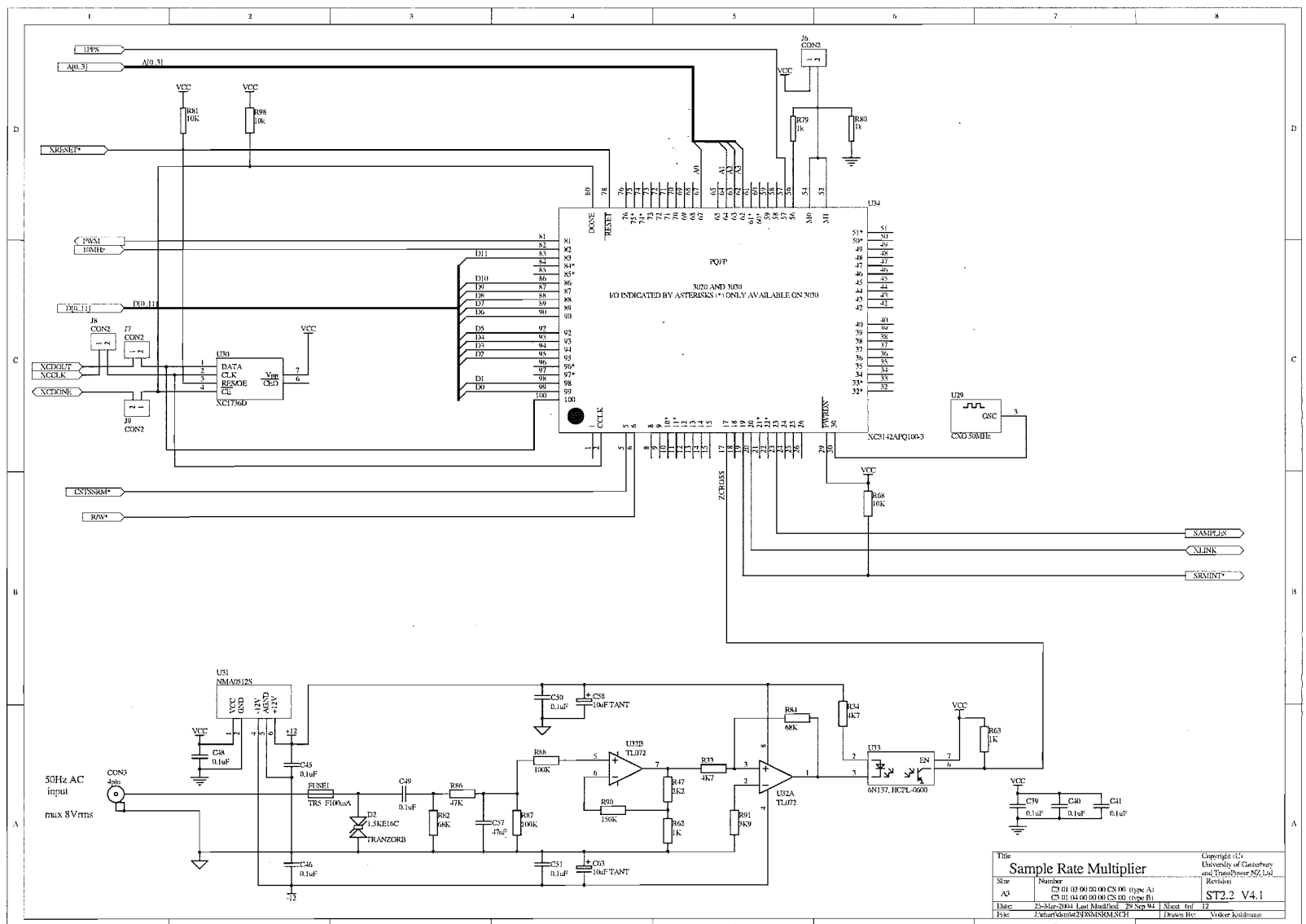

MIX
DSPM
MC
P2
U9
U7
VCC
MC74HC367
PALCE16V8
S72.2 V3.1
DSM
Volker Kuhlmann
M.J.Hurst, Vk.

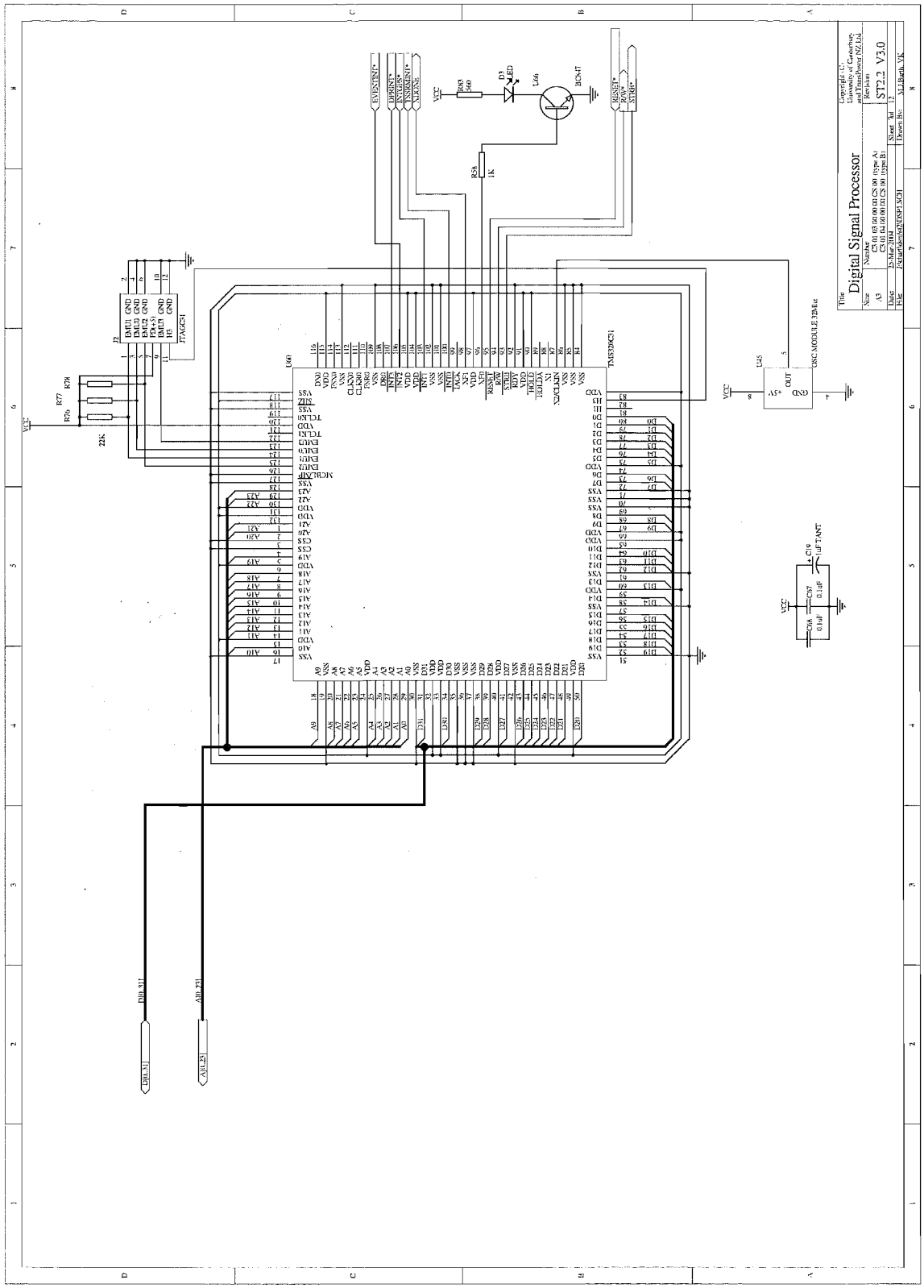

DSP
DSP1.SCH
RESET*
TSSRMINT*
INTGPS*
EVENTINT*
XDONE
D[0..31]
A[0..23]
R/W*
STRB*
DPRINT*
RMM1.SCH
CERAM*
R/W*
A[0..23]
D[0..31]
Ram Memory Module
D[0..31]
RESET*
VCC
R89
22K
DPR1
DPR1.SCH
R/W*
CEDPRL*
DPRINT*
D[0..31]
A[0..12]
SEML*
SEMR*
MEMRW*
CEDPRR*[0..3]
MEMINT*
MEMD[0..31]
MEMA[2..14]
MEMRW*
CEDPRR*[0..3]
MEMINT*
MEMD[0..31]
MEMA[2..14]
SEMR*
DSP
U24
A4
A5
A13
A14
A15
A16
A17
A18
A19
A23
CLK
OE
PAL22V10
O1
O2
O3
O4
O5
O6
O7
O8
O9
O10
DSMPARIO
DSMPARIO.SCH
D[0..7]
IORD*
IOWR*
FOEVENT*[1..3]
EVENT[1..3]
DSMFOGPS
DSMFOGPS.SCH
FOEVENT*[1..3]
D[0..7]
INTGPS*
A[0..3]
WRGPS*
RDGPS*
RESET*
1PPS
XCTRL
DSMTS
DSMTS.SCH
A[0..3]
D[0..15]
TSSRMINT*
CSTSSRM*
EVENTINT*
R/W*
1PPS
XDONE
XCLK
XCTRL
EVENT[1..3]
EVENT*
TS[0..31]
XLINK
SRMINT*
10MHz
PWM
XCHNDONE
XCHNCLK
XCHNDOUT
10MHz
DSMSRM
DSMSRM.SCH
D[0..11]
R/W*
CSTSSRM*
A[0..3]
1PPS
XCHNDOUT
XCHNCLK
XCHNDONE
PWM
10MHz
SRMINT*
XLINK
SAMPLES
SAMPLES
VCC
C47
0.1uF
Title
DSP Module
Copyright (C)
University of Canterbury
and TransPower NZ Ltd
Size
A3
Number
C5 01 03 00 00 00 C5 00 (type A)
C5 01 04 00 00 00 C5 00 (type B)
Revision
ST2.2 V3.1
Date:
25-May-2004
Sheet 2of 12
File:
P:\chart\dsm\st2\DSP\M1.SCH
Drawn By:
M.J.Barth, VK

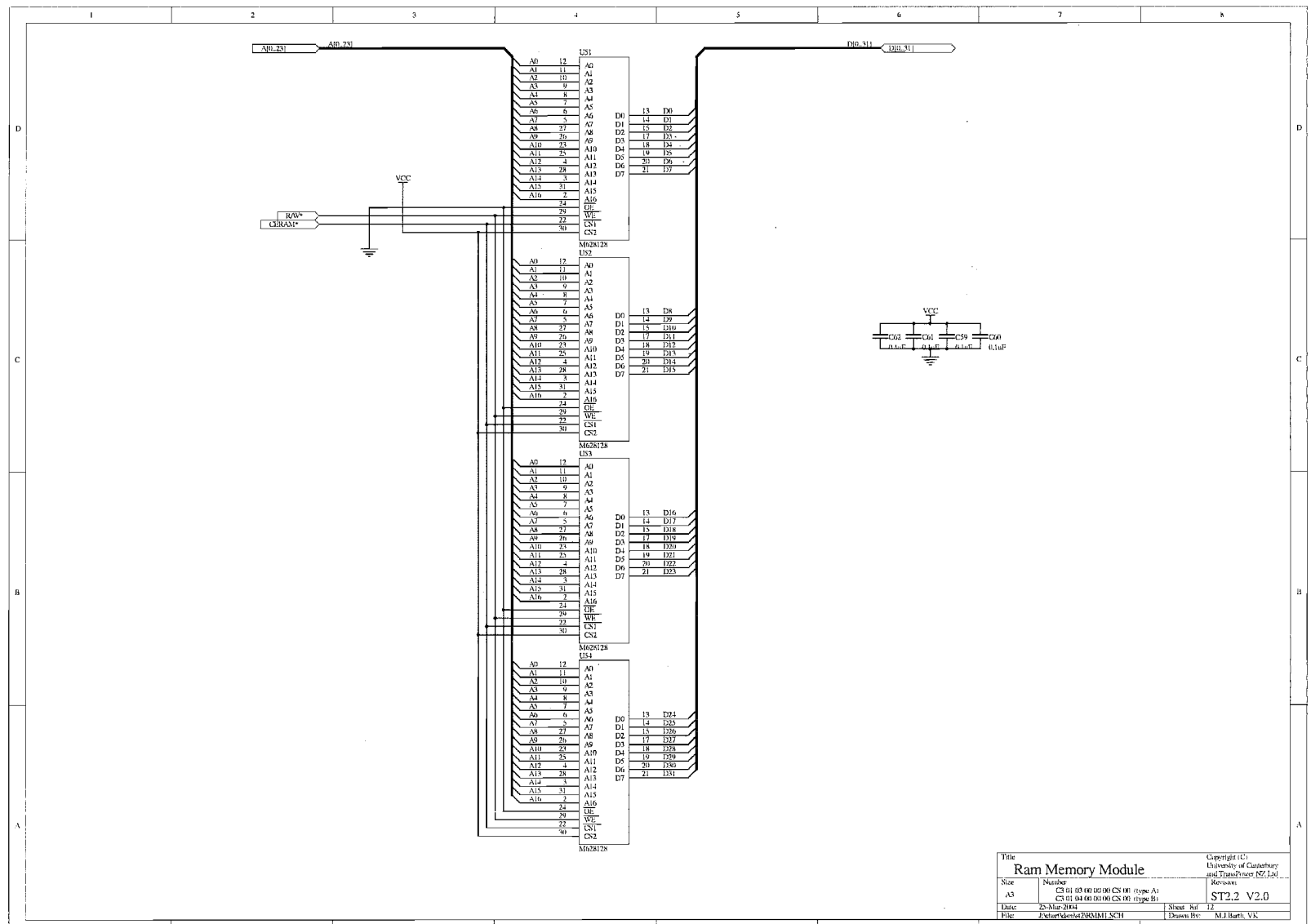

F/O and GPS Interfaces
ST2.2 V3.3
Copyright (C) University of Canterbury and Trimble Navigation
Kolbeinsson
Volker
D9 male print
CON4
D9 female print
D9 RS232-C:
2 RxD
3 TxD
5 Gnd
D25 RS232-C:
2 TxD
3 RxD
7 Gnd
The next resistor should have this val.
FO results/strobes/readyes are currently AC coupled!
=> Should be DC (as FIG4)
SCC74694
MC74HC4053
DS75176B
74HC00
74HC30
74HC4047
ODL50-126AAC
LSF MGTE 1X
JMP2
SW-PUSH
LED
RESET*
INTERR*
IRPS
A0..3]
D0..7]
FOEVENT*1..3]
FOEVENT*1..3E]
FOIN1
FOIN2
FOIN3
RxDA
TxDA

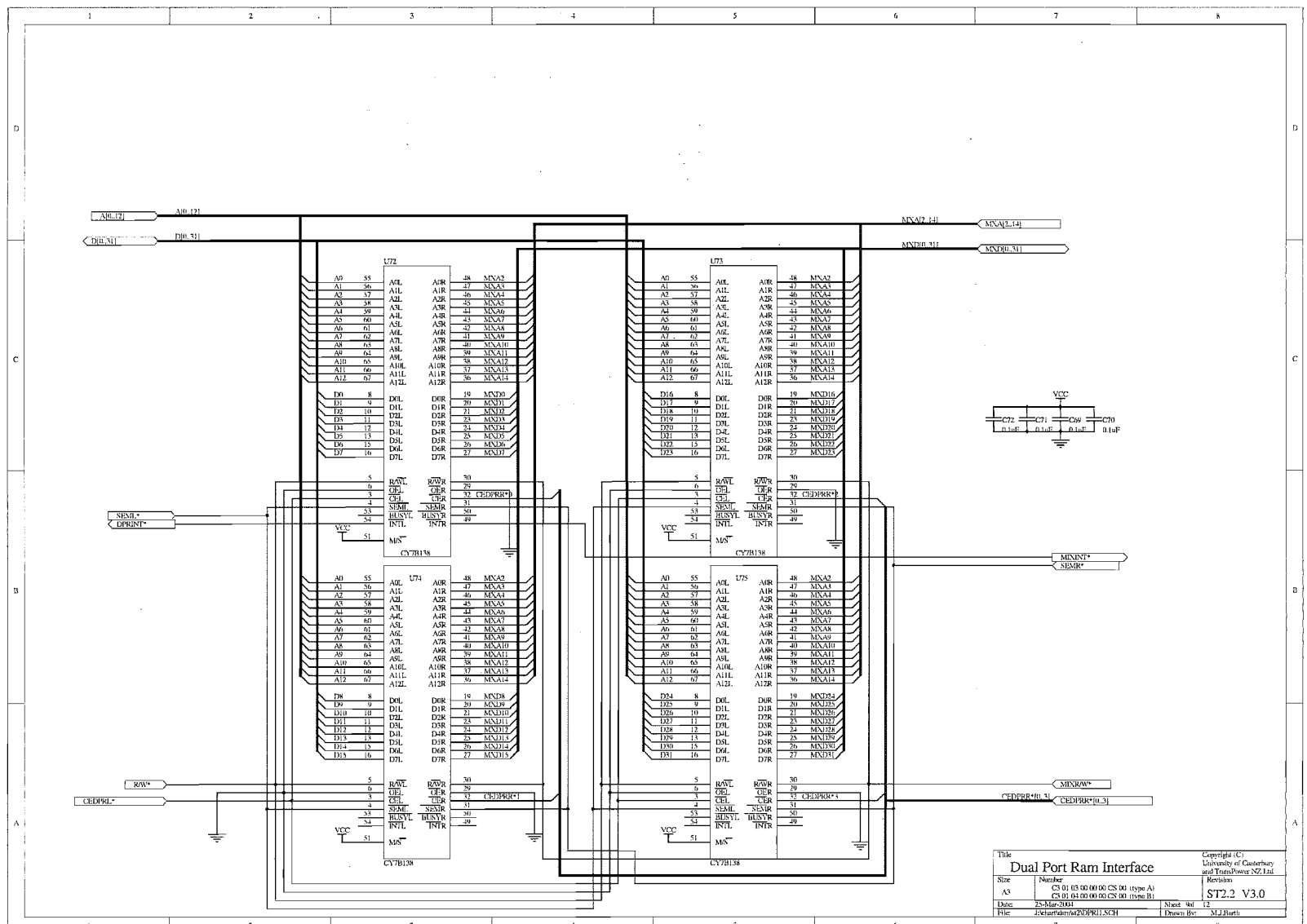

CON1
DB44 female petp
U18
U17
U19
TIL193
R59
R57
R56
R55
150
R51
R50
R49
R48
330
R38
R35
R37
R39
2K2
VCC
U23
MC74HC245A
U22
MC74HC574
IORD*
IOWR*
D[0..7]
R54
R53
R52
R60
J3
JMP2
J1
C32
C31
C30
0.1uF
R40
R41
R42
U14
U15
U16
74OL6010
R45
R44
R43
FOEVENT*[1..3]
EVENT[1..3]
FOEVENT*1
FOEVENT*2
FOEVENT*3
U20A
U20B
U20C
U20D
74HC00
EVENT1
EVENT2
EVENT3
C25
C26
C27
C28
C29
Parallel I/O Interface
Copyright (C)
University of Canterbury
and TransPower NZ Ltd
Size
A3
Number
C3 01 03 00 00 00 C8 00 (type A)
C3 01 04 00 00 00 C8 00 (type B)
Revision
ST2.2 V2.1
Date: 25-Mar-2004 Last Modified: 08 Jun 94
Sheet 4 of 12
File: J:\chart\dsm\st2\DSMPARIO.SCH
Drawn By: Volker Kuhlmann

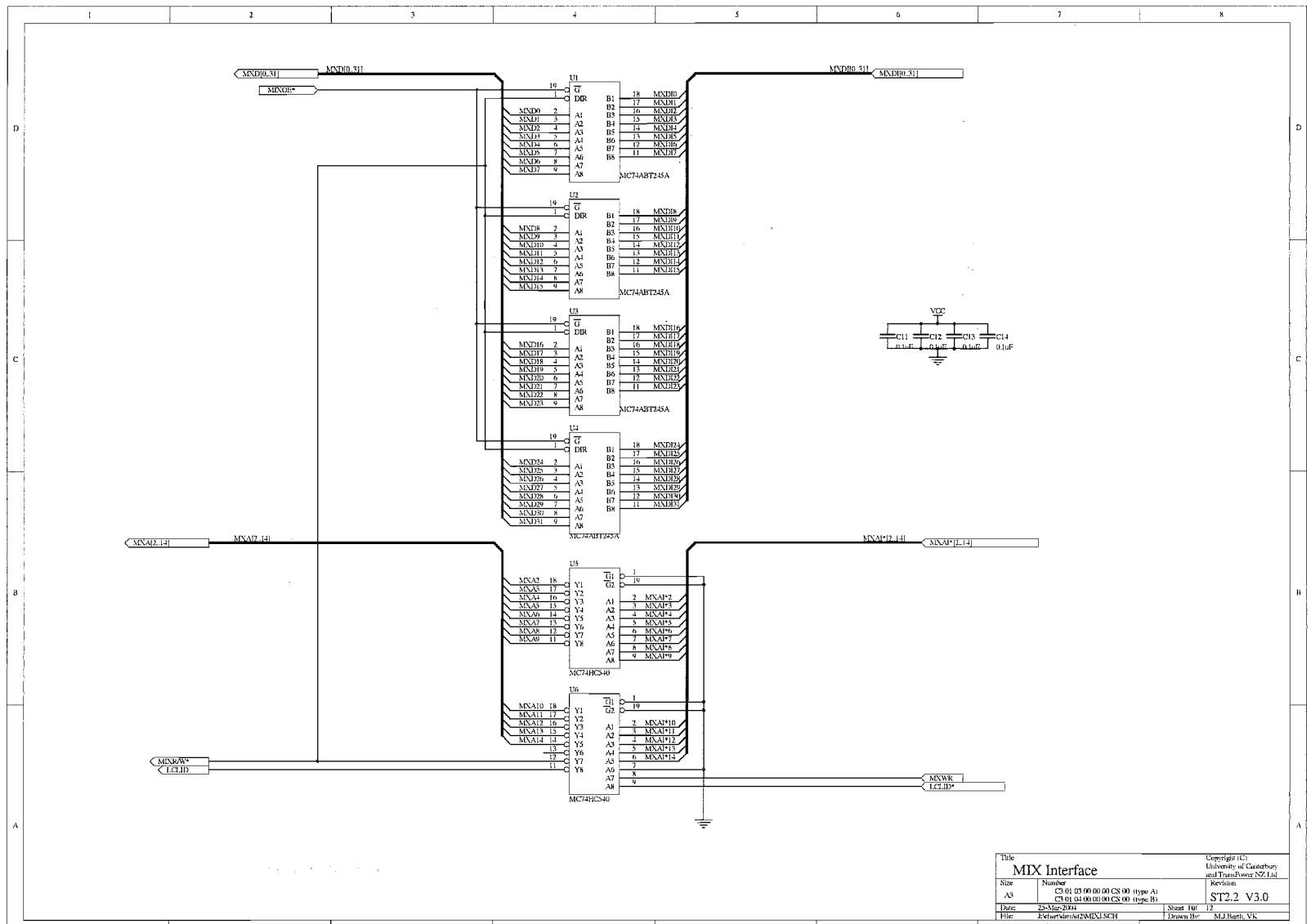

NOTES:
1) RES/OE of the XC1765D must be programmed with an active-low reset
2) The following pins are outputs during configuration: INIT*, RDY/BUSY*, DOUT, LDC*, HDC
Time Stamping
ST2.2 V3.2
Copyright (C) University of Canterbury and FlatSheet NZ Ltd.

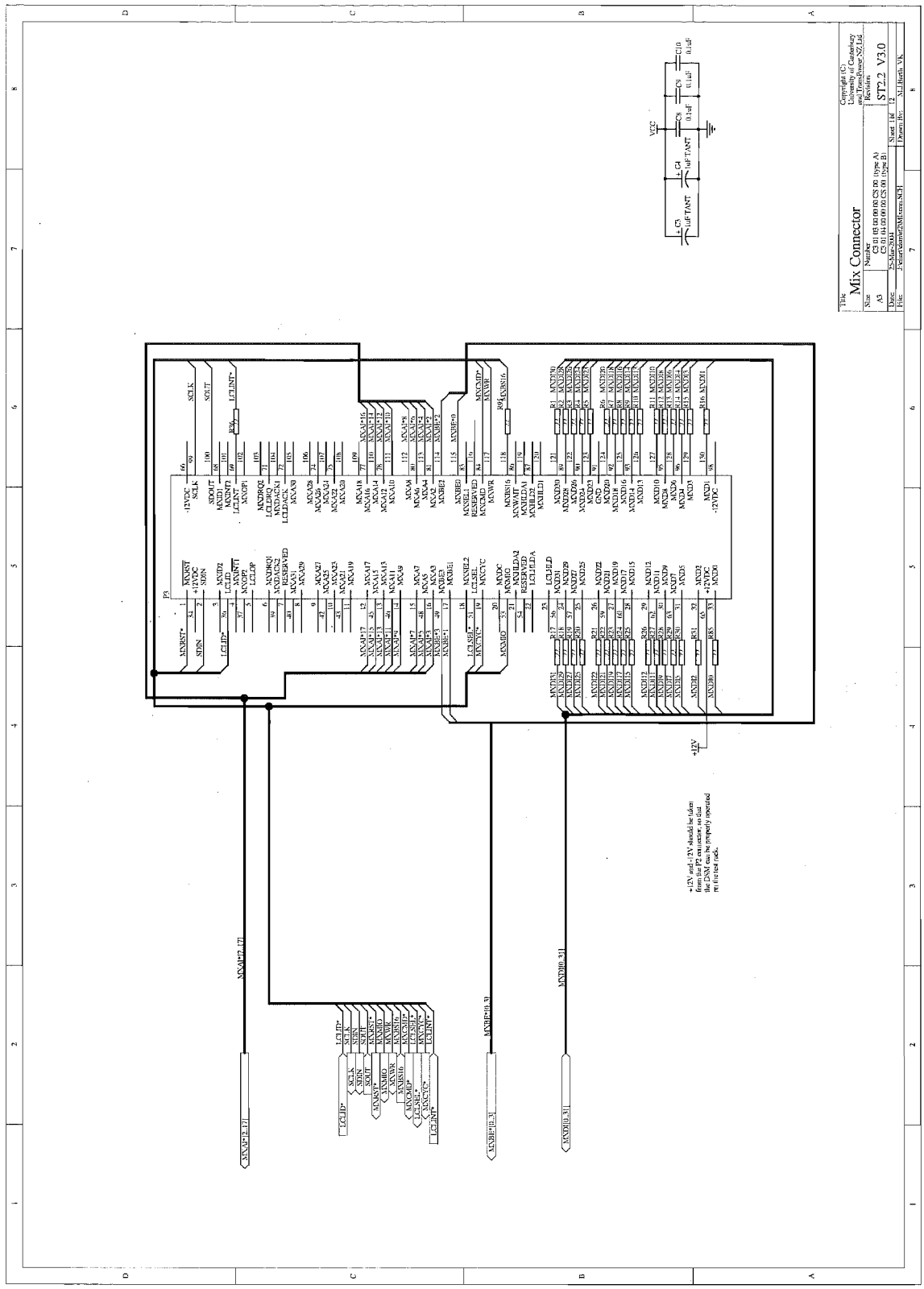

1PPS
A[0..3]
XRESET*
PWM
10MHz
D[0..11]
XCDOUT
XCCLK
XCDONE
CNTSNRM*
RAW*
SAMPLES
XLINK
SRMINT*
ZCROSS
50Hz AC input
max 8Vrms
XC3142APQ100-3
XC1736D
6N137, HCPL-0600
TRANZORB
Sample Rate Multiplier
ST2.2 V4.1
Volker Kuhlmann

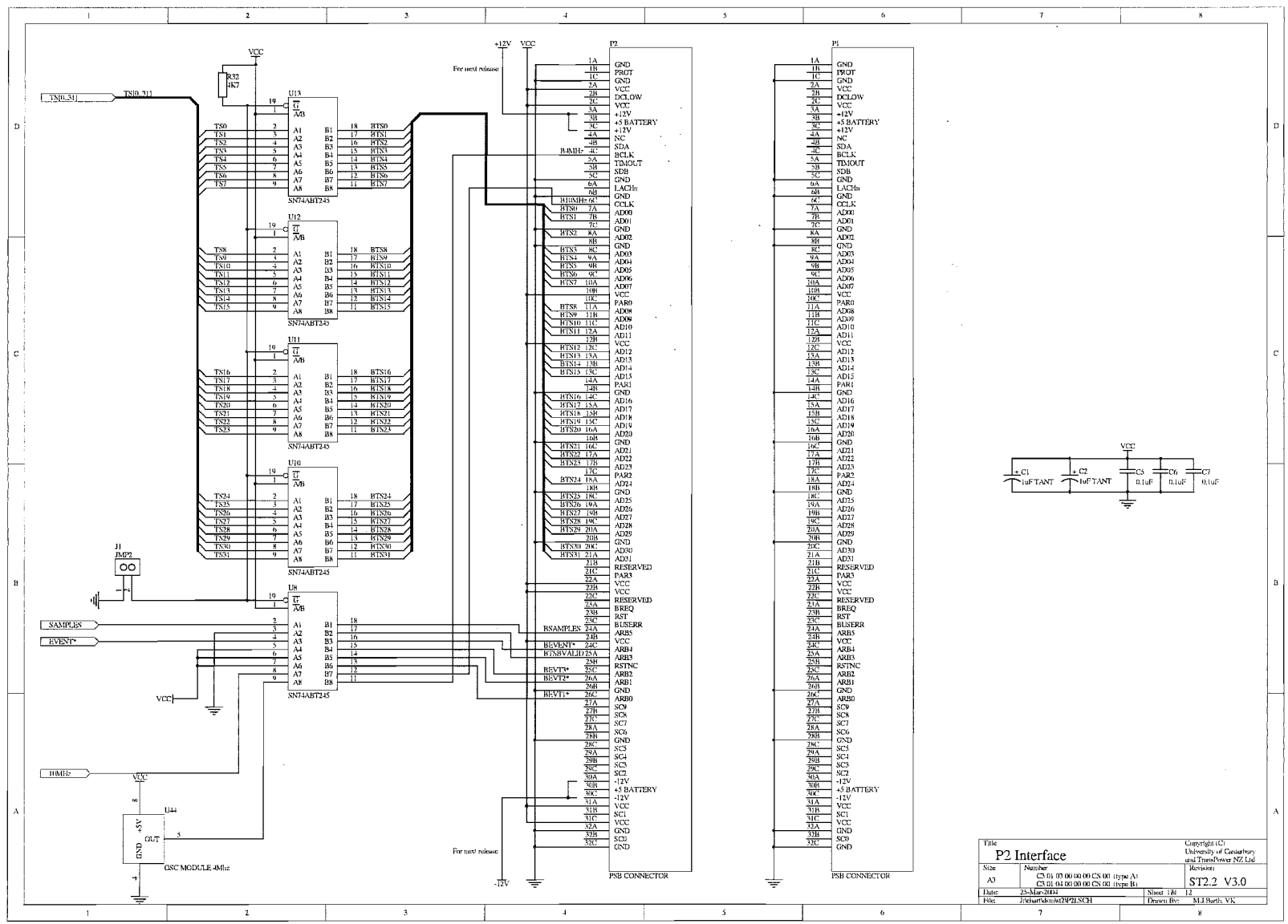

Digital Signal Processor
ST2.2 V3.0
TMS320AC31
OSC MODULE 32MHz
JTAGCH
BCW47
LED
1K
22K
VCC
EVBSYSINT*
DSPINT*
INT1/2/3*
TINP0/1/2*
XFSR0*
RESET*
BOOT*
STRB0*
D[0..31]
A[0..23]

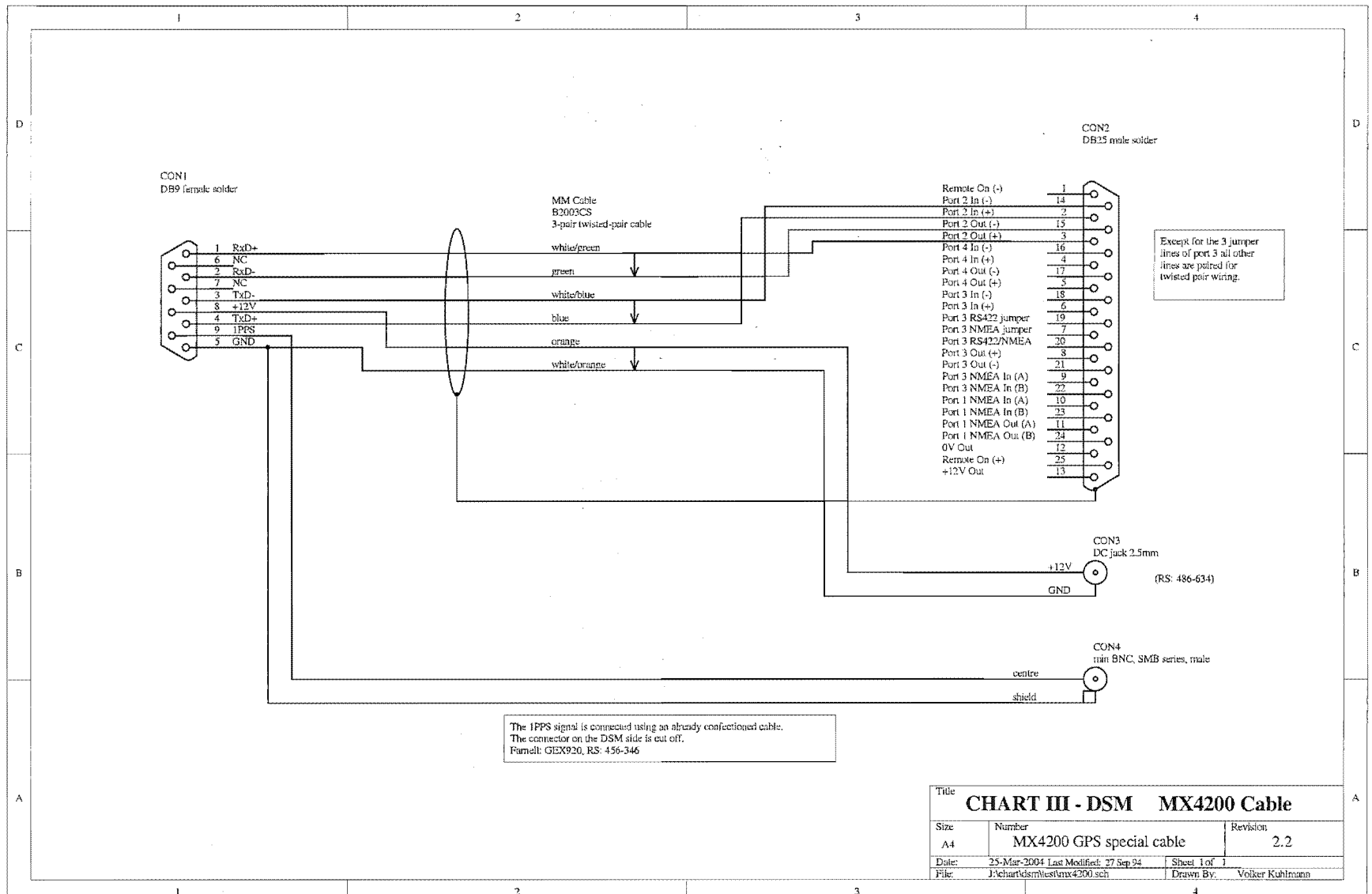

A[0..23]
D[0..31]
RAM*
CERAM*
VCC
U51
U52
U53
U54
M628128
C62 0.1uF
C61 0.1uF
C59 0.1uF
C60 0.1uF
Title
Ram Memory Module
Copyright (C) University of Canterbury and TransPower NZ Ltd
Size A3
Number C3 01 03 00 00 00 C8 00 (type A) C3 01 04 00 00 00 C8 00 (type B)
Revision ST2.2 V2.0
Date: 25-Mar-2004
Sheet 8of 12
File: E:\chart\dsm\st2\RAMM1.SCH
Drawn By: M.J Barth, V.K


A[0..12]
D[0..31]
MXA[2..14]
MXD[0..31]
U72
U73
U74
U75
CY7B138
VCC
C72 0.1uF
C71 0.1uF
C69 0.1uF
C70 0.1uF
SEML*
DPRINT*
MINT*
NEMR*
R/W*
CEDPRL*
MINRW*
CEDPRR*[0..3]
Title
Dual Port Ram Interface
Copyright (C) University of Canterbury and TransPower NZ Ltd
Size
A3
Number
CS 01 03 00 00 00 CS 00 (type A)
CS 01 04 00 00 00 CS 00 (type B)
Revision
ST2.2 V3.0
Date: 25-Mar-2004
Sheet 4of 12
File: J:\start\dsm\st2\DPRI1.SCH
Drawn By: M.J.Barth


MXDI[0..31]
MXOE*
U1
U2
U3
U4
MC74ABT245A
MXDI[0..31]
MXDI[0..31]
VCC
C11 0.1uF
C12 0.1uF
C13 0.1uF
C14 0.1uF
MXA[2..14]
MXA[2..14]
MXAI*[2..14]
U5
U6
MC74HC540
MXR/W*
LCLID
MXWR
LCLID*
Title
MIX Interface
Copyright (C) University of Canterbury and TransPower NZ Ltd
Size
A3
Number
C3 01 03 00 00 00 CS 00 (type A)
C3 01 04 00 00 00 CS 00 (type B)
Revision
ST2.2 V3.0
Date: 23-Mar-2004
Sheet 10f 12
File: E:\chart\dsn\st2\MIX1.SCH
Drawn By: MJ Bath, VK


Mix Connector
ST2.2 V3.0
P3
MXRST*
+12VDC
SDIN
LCLIP*
SCLK
SDOUT
MXSMT2
LCLINT*
MXDP1
MXDP2
LCLDP
MXDMQ1
MXDACK2
RESERVED
MXA29
MXA27
MXA25
MXA23
MXA19
MXA18
MXA17
MXA15
MXA13
MXA11
MXA9
MXA7
MXA5
MXA3
MXA1
MXSEL1
LCLSEL*
MXCYC*
MXMRQ*
MXSELA2
MXSELA1
MXSELA0
RESERVED
LCLFLDA
LCLFLD
MXD31
MXD29
MXD27
MXD25
MXD23
MXD21
MXD19
MXD17
MXD15
MXD13
MXD11
MXD9
MXD7
MXD5
MXD3
MXD1
-12VDC
MXA[19:2,17]
MXSEL[2:1]
MXD[31:0]
MXDP[3:1]
+12V
+12V and -12V should be taken from PC power supply and the DSM fan be properly operated


P2 Interface
P2 PSB CONNECTOR
P1 PSB CONNECTOR
SN74ABT245
OSC MODULE 4Mhz
JMP2
For next release
Copyright (C) University of Canterbury and TransPower NZ Ltd
ST2.2 V3.0
23-Mar-2004
J:\dsm\dsm\dsm2P2.SCH
M.J.Barth; V.K.


CON1
DB9 female solder
1 RxD+
6 NC
2 RxD-
7 NC
3 TxD-
8 +12V
4 TxD+
9 1PPS
5 GND
MM Cable
B2003CS
3-pair twisted-pair cable
white/green
green
white/blue
blue
orange
white/orange
CON2
DB25 male solder
Remote On (-) 1
Port 2 In (-) 14
Port 2 In (+) 2
Port 2 Out (-) 15
Port 2 Out (+) 3
Port 4 In (-) 16
Port 4 In (+) 4
Port 4 Out (-) 17
Port 4 Out (+) 5
Port 3 In (-) 18
Port 3 In (+) 6
Port 3 RS422 jumper 19
Port 3 NMEA jumper 7
Port 3 RS422/NMEA 20
Port 3 Out (+) 8
Port 3 Out (-) 21
Port 3 NMEA In (A) 9
Port 3 NMEA In (B) 22
Port 1 NMEA In (A) 10
Port 1 NMEA In (B) 23
Port 1 NMEA Out (A) 11
Port 1 NMEA Out (B) 24
0V Out 12
Remote On (+) 25
+12V Out 13
Except for the 3 jumper lines of port 3 all other lines are paired for twisted pair wiring.
CON3
DC jack 2.5mm
+12V
GND
(RS: 486-634)
CON4
min BNC, SMB series, male
centre
shield
The 1PPS signal is connected using an already confectioned cable.
The connector on the DSM side is cut off.
Farnell: GEX920, RS: 456-346
Title
CHART III - DSM MX4200 Cable
Size A4
Number MX4200 GPS special cable
Revision 2.2
Date: 25-Mar-2004 Last Modified: 27 Sep 94
Sheet 1 of 1
File: J:\chart\dsm\test\mx4200.sch
Drawn By: Volker Kuhlmann

C.2 Connector P1 and P2 (TSB)

The pinouts of the Multibus II P1 connector, and the P2 connector carrying the TSB (Time Stamping Bus) in CHART, are given in table C.1. The DSM jumpers are given in table C.2, the DSM memory map is given in table C.2, and the DSM's FPGA registers are given in table C.4. The FPGA registers are the programming interface to the DSM hardware.

	A	B	C
1	GND	PROT	GND
2	Vcc	DCLOW	Vcc
3	+12V	+5VBatt	+12V
4	NC	SDA	BCLK
5	TIMOUT	SDB	GND
6	LACHn	GND	CCLK
7	AD00	AD01	GND
8	AD02	GND	AD03
9	AD04	AD05	AD06
10	AD07	Vcc	PAR0
11	AD08	AD09	AD10
12	AD11	Vcc	AD12
13	AD13	AD14	AD15
14	PAR1	GND	AD16
15	AD17	AD18	AD19
16	AD20	GND	AD21
17	AD22	AD23	PAR2
18	AD24	GND	AD25
19	AD26	AD27	AD28
20	AD29	GND	AD30
21	AD31	reserved	PAR3
22	Vcc	Vcc	reserved
23	BREQ	RST	BUSERR
24	ARB5	Vcc	ARB4
25	ARB3	RSTNC	ARB2
26	ARB1	GND	ARB0
27	SC9	SC8	SC7
28	SC6	GND	SC5
29	SC4	SC3	SC2
30	−12V	+5VBatt	−12V
31	Vcc	SC1	Vcc
32	GND	SC0	GND

P1 connector pinout

	A	B	C
1	GND		GND
2	Vcc		Vcc
3	+12V	+5VBatt	+12V
4			4MHz
5			GND
6		GND	6MHz
7	TS0	TS1	GND
8	TS2	GND	TS3
9	TS4	TS5	TS6
10	TS7	Vcc	
11	TS8	TS9	TS10
12	TS11	Vcc	TS12
13	TS13	TS14	TS15
14		GND	TS16
15	TS17	TS18	TS19
16	TS20	GND	TS21
17	TS22	TS23	
18	TS24	GND	TS25
19	TS26	TS27	TS28
20	TS29	GND	TS30
21	TS31		
22	Vcc	Vcc	
23			
24	SAMPLES	Vcc	EVENT*
25	TSBVALID		EVT3*
26	EVT2*	GND	EVT1*
27			
28		GND	
29			
30	−12V	+5VBatt	−12V
31	Vcc		Vcc
32	GND		GND

P2 (TSB) connector pinout

Table C.1: Pinout of the P1 and P2 Multibus II backplane connectors.

Jumper	Shipped	Function
J1	X	Enables the TSB drivers when closed.
J3	—	Supplies Vcc to the PARIO connector when closed.
J4	—	Supplies GND to the PARIO connector when closed.
J5	X	XC3164A configuration loading: open: master serial mode, closed: slave serial mode.
J6, J7, J8	X	Must be closed if FPGA configuration is chain-loaded.
J9	X	XC3142A configuration loading: open: master serial mode, closed: slave serial mode.

Table C.2: The function of the jumpers used on the DSM. The column "shipped" shows the settings when the board is shipped, an "X" means closed.

Range	Length	Use
0000000-00000BF	C0	Vectors
0000000-0000FFF	1000	Dual-port RAM
0020000-003FFFF	20000	RAM
0809800-0809BFF	400	Internal DSP RAM block 0
0809C00-0809FFF	400	Internal DSP RAM block 0
0810000-081000F	10	Semaphores of the dual-port RAM
0810010-081001F	10	FPGA registers
0810020-081002F	10	DUART registers
0810030	1	PARIO
0818000	1	FPGA

Table C.3: Memory map of the DSM. All addresses and lengths are given in hex. The lengths are in words of 32 bit.

Address		Write	Read
0	0x00	Clear event 1 interrupt flag	Interrupt flags
1	0x01	Clear event 2 interrupt flag	—
2	0x02	Clear event 3 interrupt flag	—
3	0x03	Clear TC interrupt flag	—
4	0x04	Clear 1PPS interrupt flag	—
5	0x05	Clear zero-crossing interrupt flag	—
6	0x06	—	—
7	0x07	Clear all interrupt flags	—
8	0x08	DSM configuration register	Event 1 vernier 15–0
9	0x09	SRM division register	Event 2 vernier 15–0
10	0x0A	TSB bits 15–0	Event 3 vernier 15–0
11	0x0B	TSB bits 31–16	15–8: event 2 vernier 23–16, 7–0: event 1 vernier 23–16
12	0x0C	PWM	Event 3 vernier 23–16
13	0x0D	—	Vernier correction
14	0x0E	—	SRM correction
15	0x0F	—	—

Table C.4: The registers implemented in the FPGAs.

appendix D

The GAL Listings

D.1 MIX Address Decoder

```
""
Title:          CHART III DSM

                University of Canterbury
                Dept of Electrical and Electronic Engineering
                Creyke Rd

                CHART III Project
                Mr M.B. Dewe (project supervisor)

Copyright (C):  University of Canterbury and TransPower NZ Ltd

Design:         Address Decoder for DSM ST2 (C31) MIX Interface
Drawing No:     C3 01 03 00 03 XX PI 00
                C3 01 04 00 03 XX PI 00

Device :        GAL22V10

Author:         Mike Hodkin, Volker Kuhlmann
Assembler:      PLANII version 1.10 (National Semiconductor)
File:           DSM31MIX.EQN
Release:        ST2
Version:        1.2
Date:           31 August 1994
""


CHIP DSM_MIX_AddressDecoder GAL22V10


;PINLIST

;pin   1       2       3    4        5       6    7       8
       /MXCMD  /MXRST MXA7 /LCLSEL  /MXCYC MXWR /MXBE0 /MXBE1

;               9       10     11     12
                /MXBE2  /MXBE3 MXD0   GND

;       13    14      15       16       17       18       19
       MXMIO /MXINT /RESET /CEDPRR0 /CEDPRR1 /CEDPRR2 /CEDPRR3

;               20      21     22      23     24
                /MDPSEM /MIXOE /LCLINT LCLID  VCC


@UES    DSMMX1.2
```

```
EQUATIONS

LCLID    = GND
LCLID.OE = MXRST
;tristate enabled during MIX reset

LCLINT    = MXINT
LCLINT.OE = VCC
;only buffered

/RESET     := /RESET
.SETF      = LCLSEL & MXCMD & /MXMIO & MXA7 & MXWR & /MXD0
.RSTF      = LCLSEL & MXCMD & /MXMIO & MXA7 & MXWR & MXD0
;After powerup all FF are low. FF low means output high if output is inverted!
;Now powering up with /RESET = low.
;Make /RESET low when /MXRST is low?

CEDPRR0    =  LCLSEL & MXCYC & /MXRST & MXMIO & MXBE0
CEDPRR0.OE = VCC
;DPRAMlow enabled only for data read/write of 8/16 bits, timing needs MXCYC

CEDPRR1    =  LCLSEL & MXCYC & /MXRST & MXMIO & MXBE1
CEDPRR1.OE = VCC
;DPRAMlow enabled only for data read/write of 8/16 bits, timing needs MXCYC

CEDPRR2    =  LCLSEL & MXCYC & /MXRST & MXMIO & MXBE2
CEDPRR2.OE = VCC
;DPRAMlow enabled only for data read/write of 8/16 bits, timing needs MXCYC

CEDPRR3    =  LCLSEL & MXCYC & /MXRST & MXMIO & MXBE3
CEDPRR3.OE = VCC
;DPRAMlow enabled only for data read/write of 8/16 bits, timing needs MXCYC

MIXOE      = LCLSEL & MXCMD & /MXRST
MIXOE.OE   = VCC
;DPRAM output enable timing from MXCMD during any selected cycle.

MDPSEM     = LCLSEL & MXCYC & /MXRST & /MXMIO & /MXA7
MDPSEM.OE  = VCC
;DPRAM semaphore write, only during i/o mapped access with MXA7 not set
```

D.2 DSP Address Decoder

```
""
Title:          CHART III DSM

                University of Canterbury
                Dept of Electrical and Electronic Engineering
                Creyke Rd

                CHART III Project
                Mr M.B. Dewe (project supervisor)

Copyright (C):  University of Canterbury and TransPower NZ Ltd

Design:         Address Decoder for DSP
Drawing No:     C3 01 03 00 04 XX PI 00
                C3 01 04 00 04 XX PI 00

Device:         GAL22V10

Author:         Allan Miller, Volker Kuhlmann
Assembler:      PLANII version 1.10 (National Semiconductor)
File:           DSP.EQN
Release:        ST2
Version:        3.2
```

```
Date:           28 September 1994
""


CHIP DSM_DSP_AddressDecoder GAL22V10


;PINLIST

;pin 1     2  3  4  5   6   7   8   9   10  11  12
     /STRB /W A4 A5 A13 A14 A15 A16 A17 A18 A19 gnd

;pin 13  14      15    16       17    18 19      20     21   22       23   24
     A23 /RDUART /IORD /CSTSSRM /IOWR NC /WRUART /CERAM XCLK /CEDPRAM /SEM Vcc


@UES    DSM3.2


EQUATIONS

;RAM
CERAM =   /A23 *  A17 * STRB
CERAM.OE = vcc

;Dual-Port RAM
CEDPRAM = /A23 * /A17 * /A16 * /A15 * /A14 * /A13 * STRB
CEDPRAM.OE = vcc

;Dual-Port RAM Semaphores
SEM =     A23 * /A19 * /A18 * /A17 *  A16 * /A15 * /A5 * /A4 * STRB
SEM.OE = vcc

;XILINX Registers
CSTSSRM =  A23 * /A19 * /A18 * /A17 *  A16 * /A15 * /A5 *  A4 * STRB
CSTSSRM.OE = vcc

;UART (GPS)
RDUART =   A23 * /A19 * /A18 * /A17 *  A16 * /A15 *  A5 * /A4 * /W * STRB
RDUART.OE = vcc
;
WRUART =   A23 * /A19 * /A18 * /A17 *  A16 * /A15 *  A5 * /A4 *  W * STRB
WRUART.OE = vcc

;Parallel IO
IORD =     A23 * /A19 * /A18 * /A17 *  A16 * /A15 *  A5 *  A4 * /W * STRB
IORD.OE = vcc
;
IOWR =     A23 * /A19 * /A18 * /A17 *  A16 * /A15 *  A5 *  A4 *  W * STRB
IOWR.OE = vcc

;Uploading the XILINX configuration
/XCLK =    A23 * /A19 * /A18 * /A17 *  A16 *  A15 * /A5 * /A4 *  W * STRB
XCLK.OE = vcc

;Global asynchronous set/reset
.SETF =    gnd
.RSTF =    gnd
```

appendix E

The FPGA Schematic Diagrams

An FPGA design is created by drawing multiple sheets, and by starting the design compiler with the top-level sheet. All existing sheets are automatically included by the compiler, but those circuit elements which are not used when following the connections from the top-level sheet are automatically removed from the FPGA design. This allows the re-use of common circuit elements and simplifies workflow.

The DSM makes use of this for its two FPGAs. The re-use of the address decoder by both FPGAs ensures that the address space occupied by each FPGA does not interfere with the other FPGA. From the DSM DSP programmer's point of view, both FPGAs appear as a single chip.

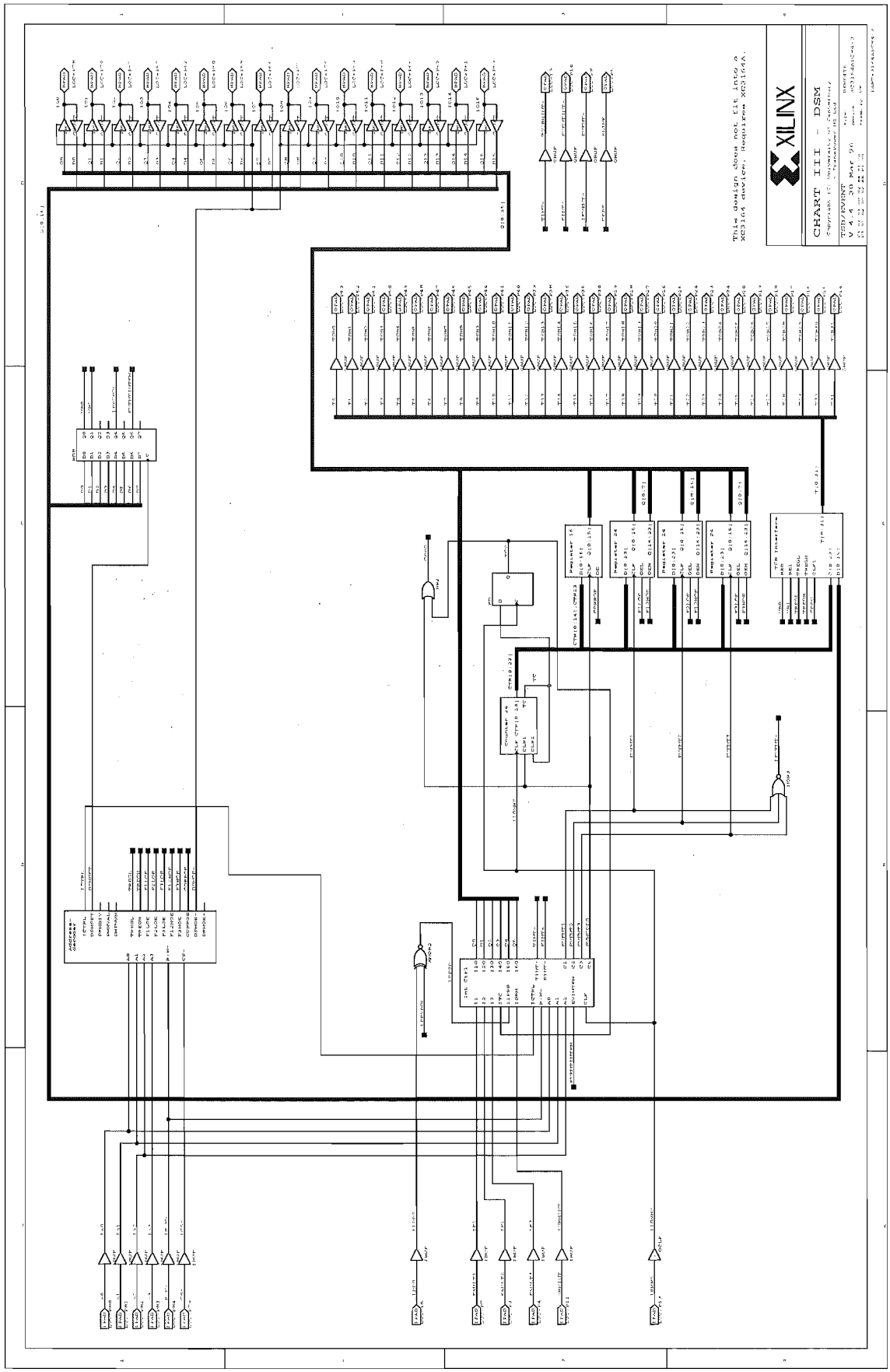
Because the DSM software allows to load different designs into both FPGAs, alternative designs are possible and more than two top-level sheets may exist. The first sheets show the top-level schematics for the:

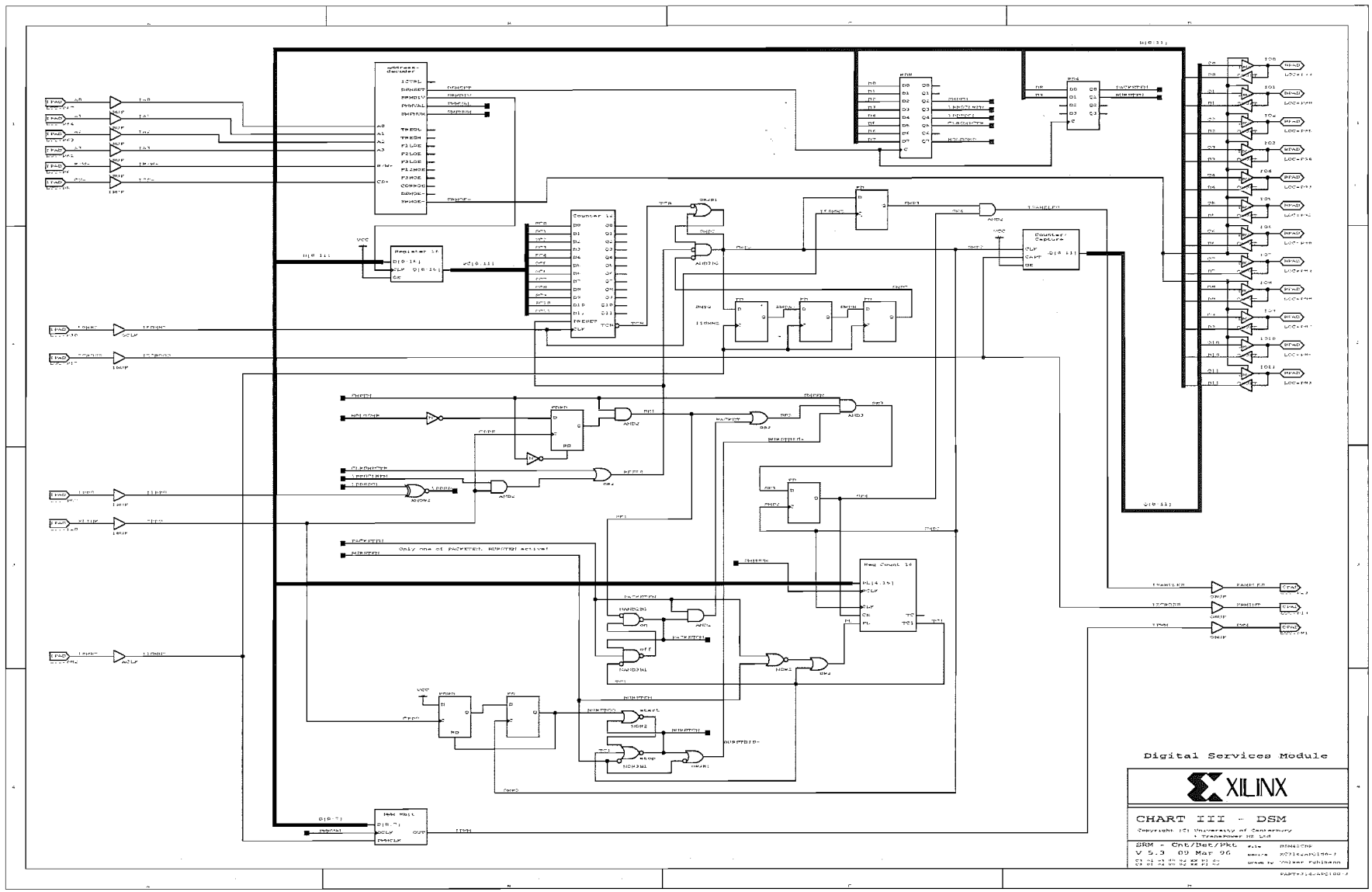
- time stamping FPGA with event capturing (page 224), the
- new SRM FPGA (page 225), and the
- alternative SRM FPGA, equivalent to the state before the changes described in chapter 8 were made (page 226).

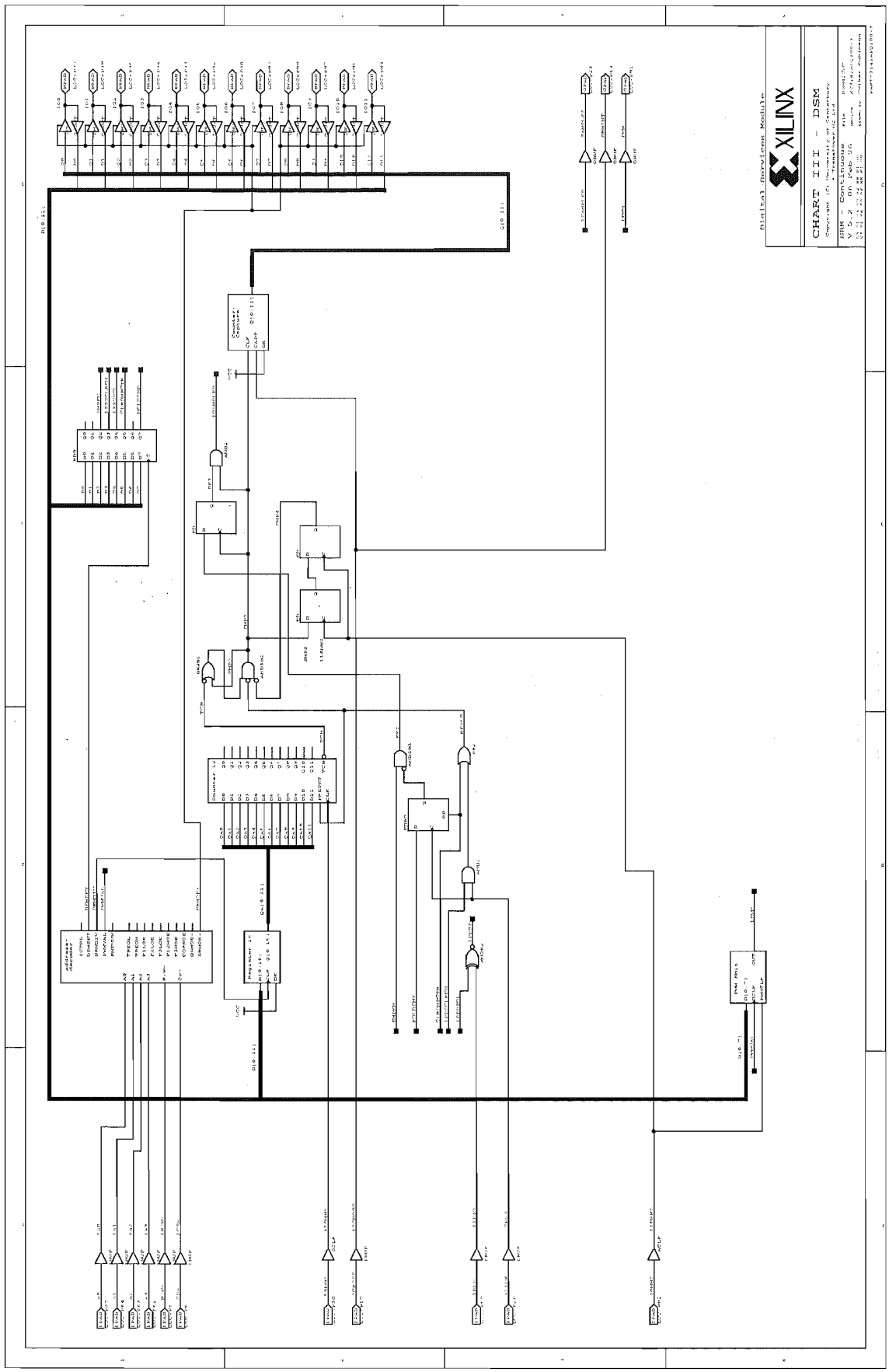
The remaining sheets are used by the top-level ones.

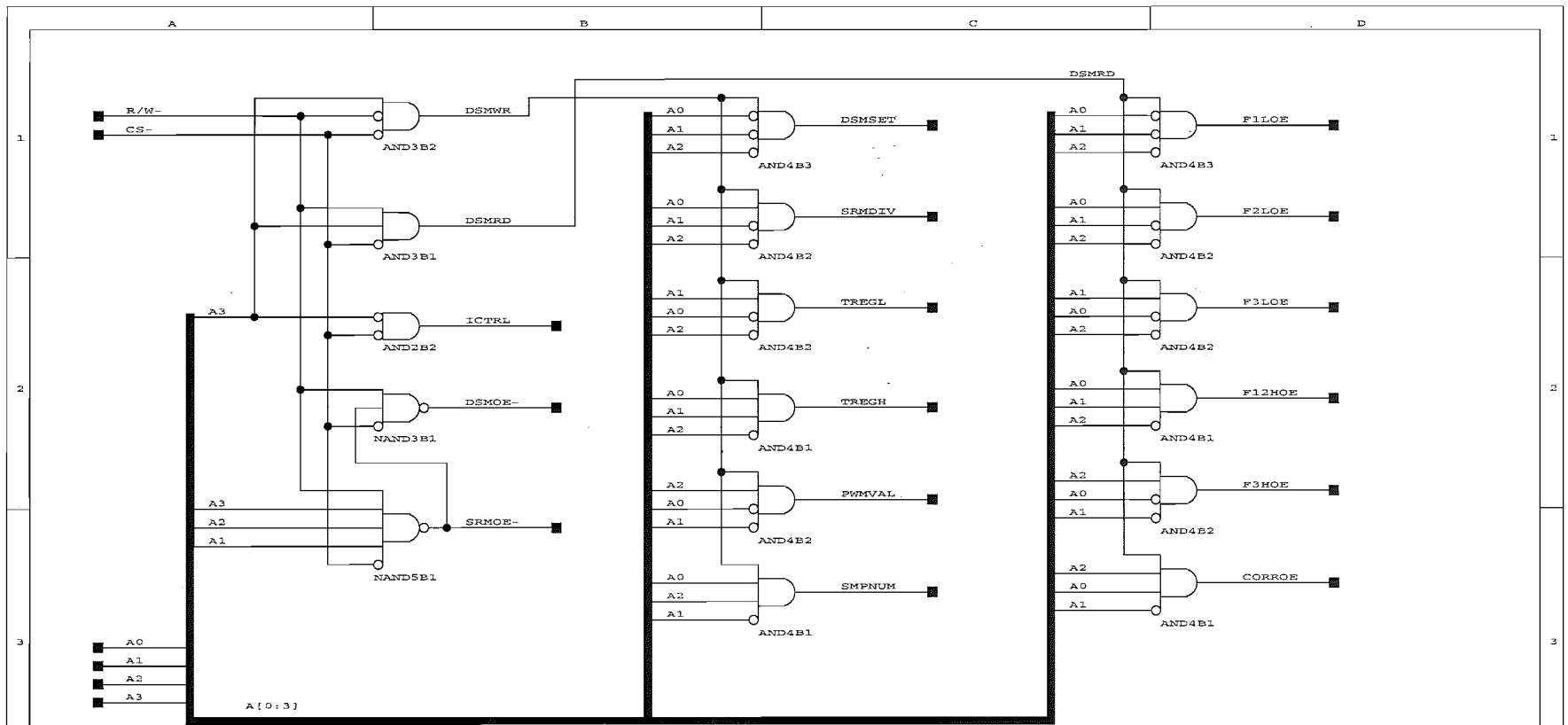
The time stamping and event capturing circuitry does not fit into an XC3164 FPGA because of the relatively high number of parallel connections required on the chip. This required an XC3195 FPGA, which is about twice the cost. With XILINX bringing the XC3100A serious to market the event capturing circuitry fits into an XC3164A because of increased routing resources.

The relevant circuitry parts for the synchronisation of the various DSM clocks are on sheets `dsm64te` and `dsm42cbp`.


XILINX
CHART III - DSM
This design does not fit into a XC3164 device, requires XC3164A.


Digital Services Module
XILINX
CHART III - DSM
SRM - Cnt/Det/Pkt
V 5.3 09 Mar 96


XILINX
CHART III - DSM



Note:
SRMOE- and DSMOE- must never be active at the same time if the design is located in 2 different chips that act as one unit.

XILINX

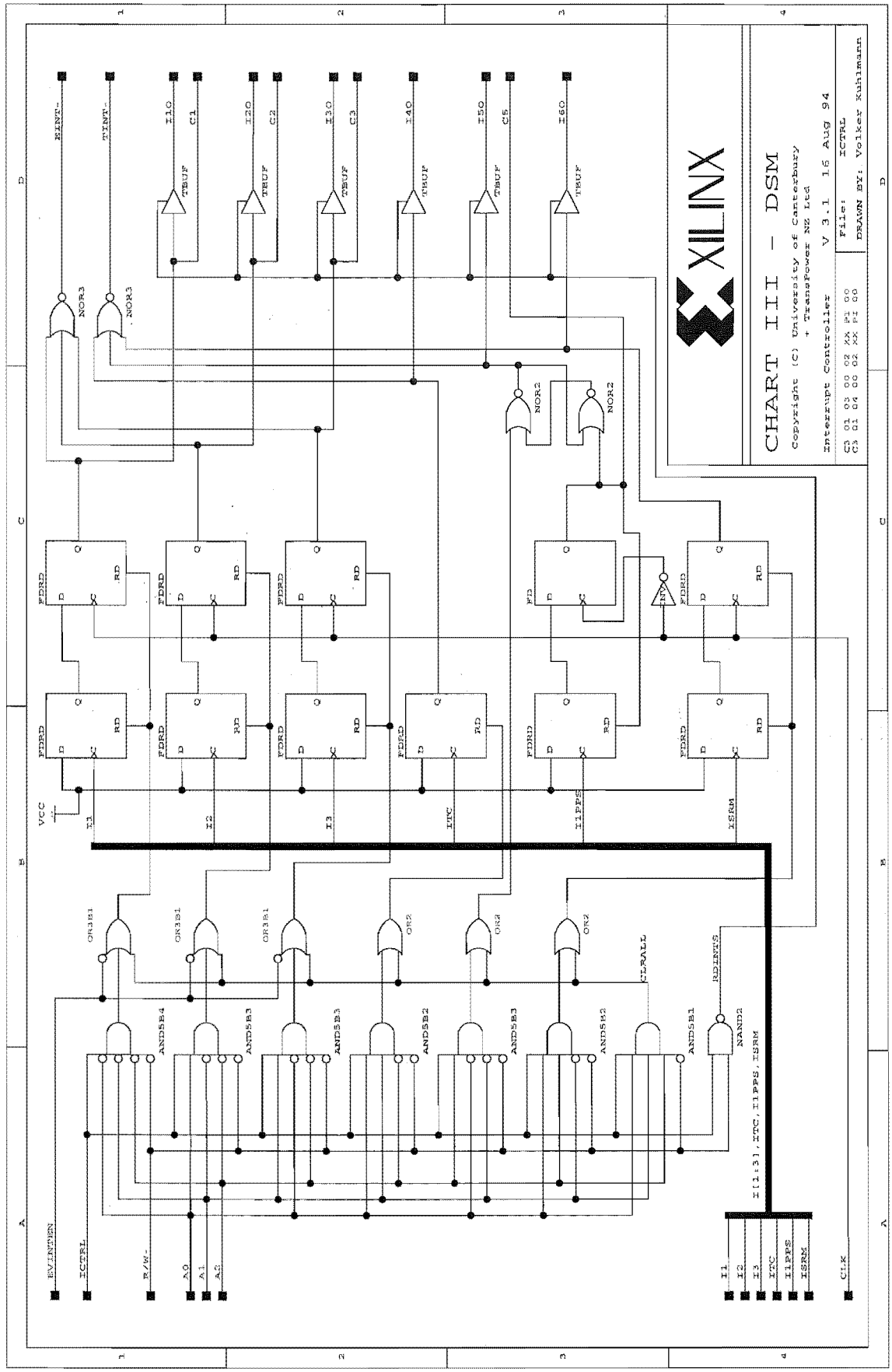
CHART III – DSM

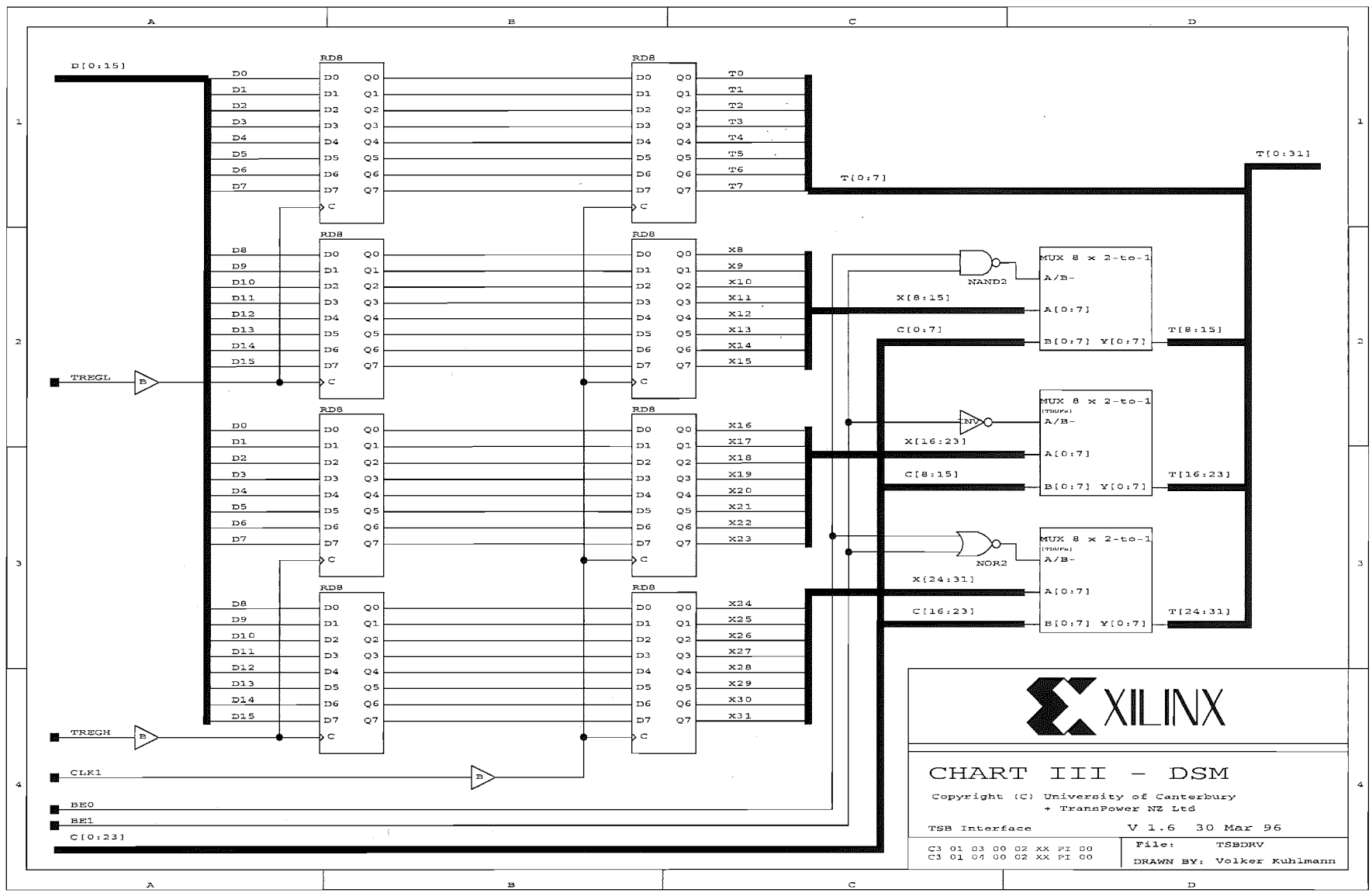
Address-Decoder V 2.3 01 Mar 96

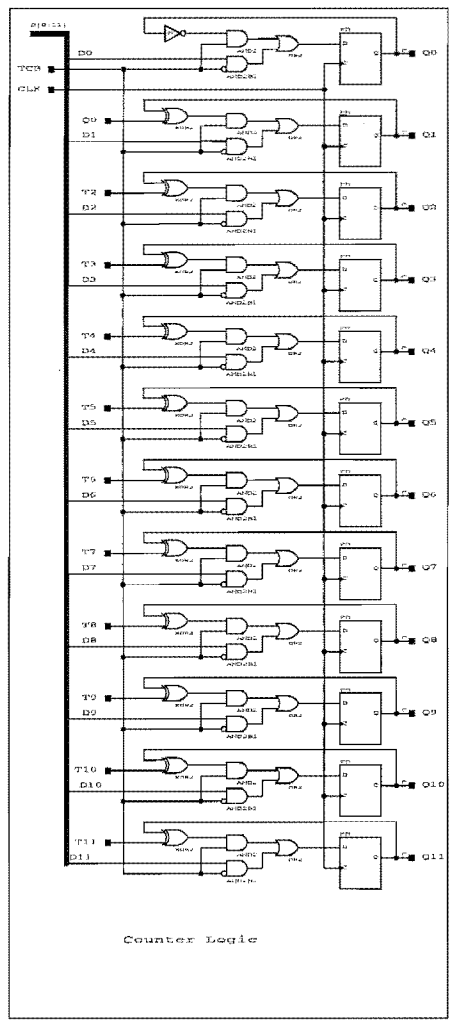
C3 01 03 00 02 XX FI 00
C3 01 04 00 02 XX FI 00

File: ADECOD

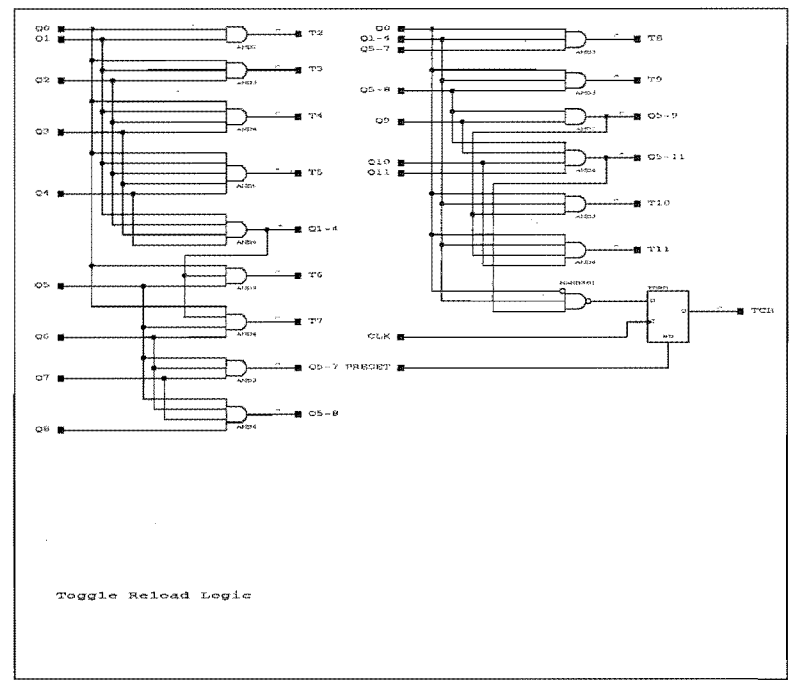
DRAWN BY: Volker Kuhlmann


XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Interrupt Controller V 3.1 16 Aug 94
File: ICTRL
DRAWN BY: Volker Kuhlmann
EINT~
TINT~
I10
C1
I20
C2
I30
C3
I40
C4
I50
C5
I60
TBUF
NOR3
NOR2
FDRD
INV
VCC
I1
I2
I3
ITC
I1PPS
ISRM
OR3B1
OR2
CLRALL
RDINTS
AND5B4
AND5B3
AND5B2
AND5B1
NAND2
I[1:3],ITC,I1PPS,ISRM
EVINTPEN
ICTRL
R/W~
A0
A1
A2
CLK


D[0:15]
RD8
TREGL
TREGH
CLK1
BE0
BE1
C[0:23]
T[0:7]
T[0:31]
X[8:15]
C[0:7]
T[8:15]
X[16:23]
C[8:15]
T[16:23]
X[24:31]
C[16:23]
T[24:31]
NAND2
INV
NOR2
MUX 8 x 2-to-1
A/B-
A[0:7]
B[0:7]
Y[0:7]
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
TSB Interface
V 1.6 30 Mar 96
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: TSBDRV
DRAWN BY: Volker Kuhlmann



Counter Logic



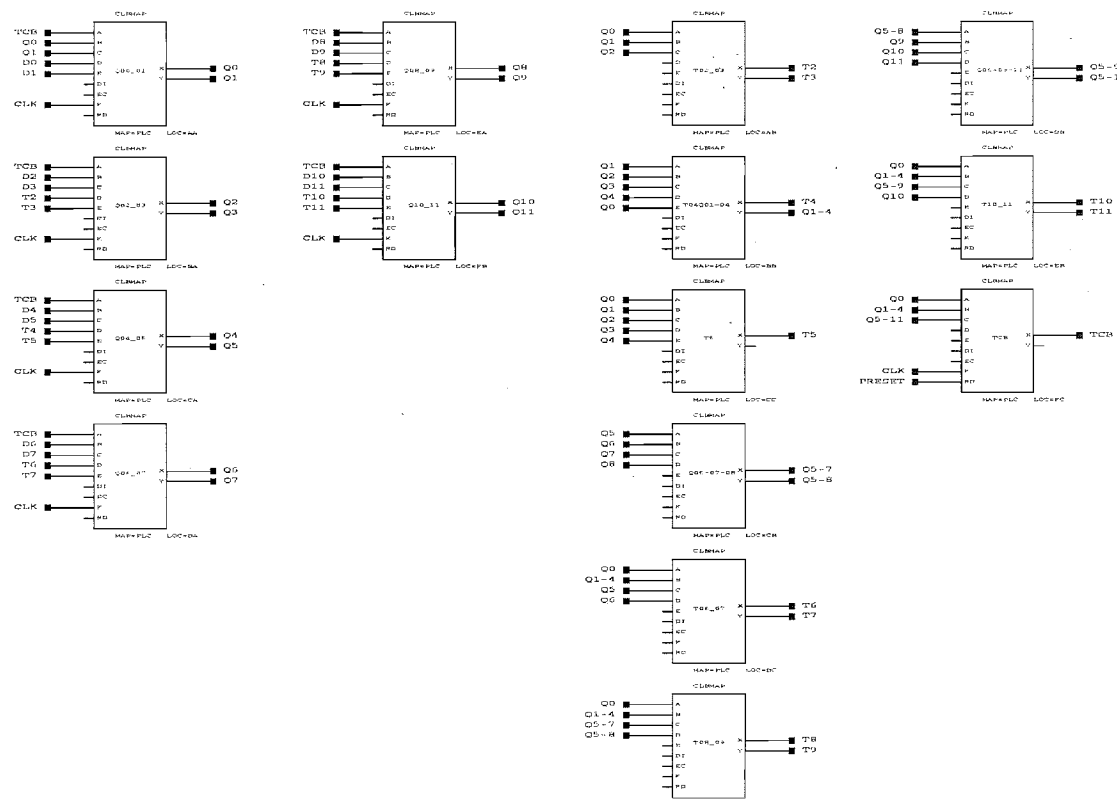
Toggle Reload Logic

XILINX

CHART III - DSM

Counter 12bit

Filename: COUNT12 (From COUNT2)
Sheet: 1 of 2
DRAWN BY: MK-01-09-93, VK on Sep 94

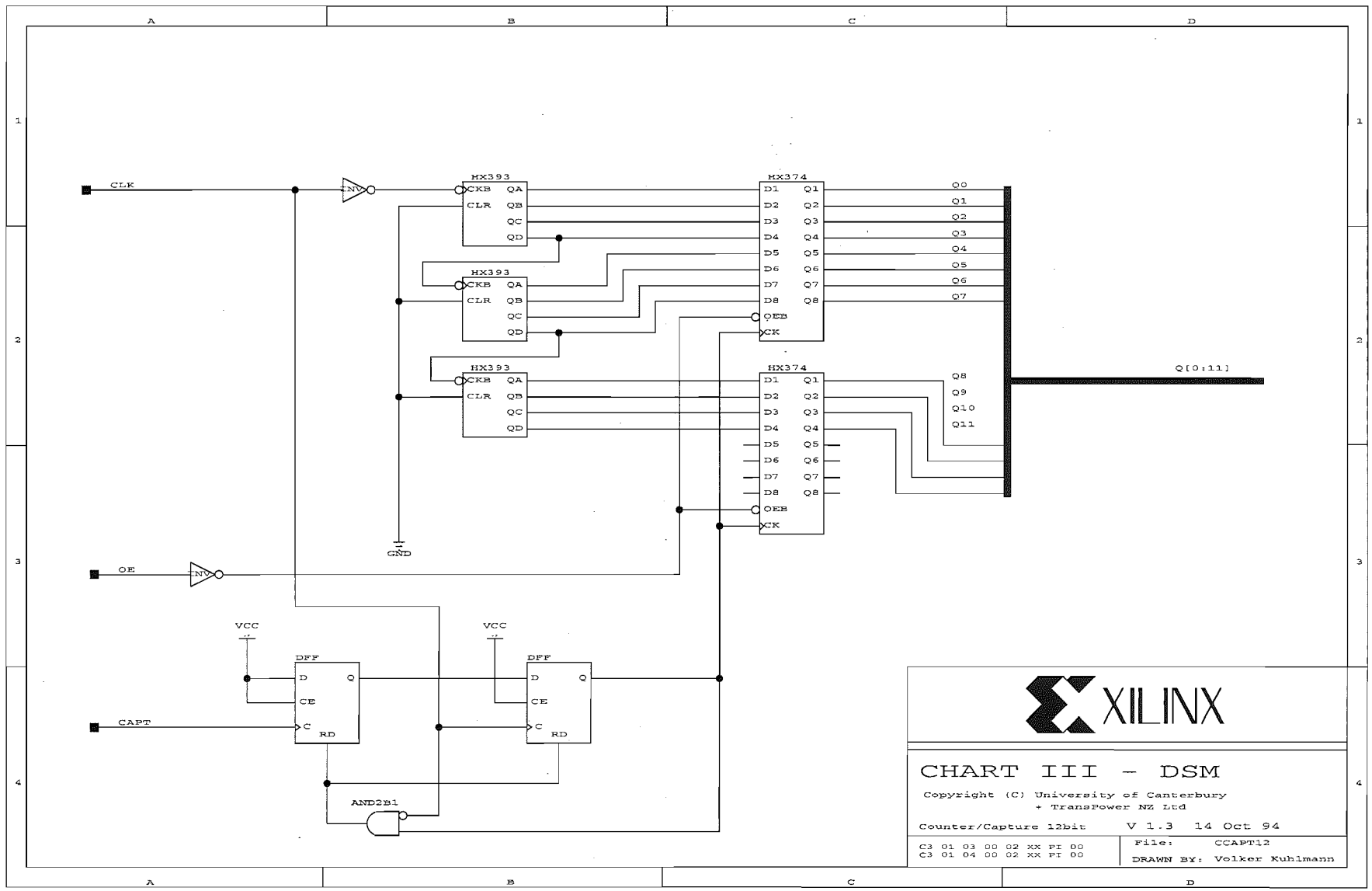


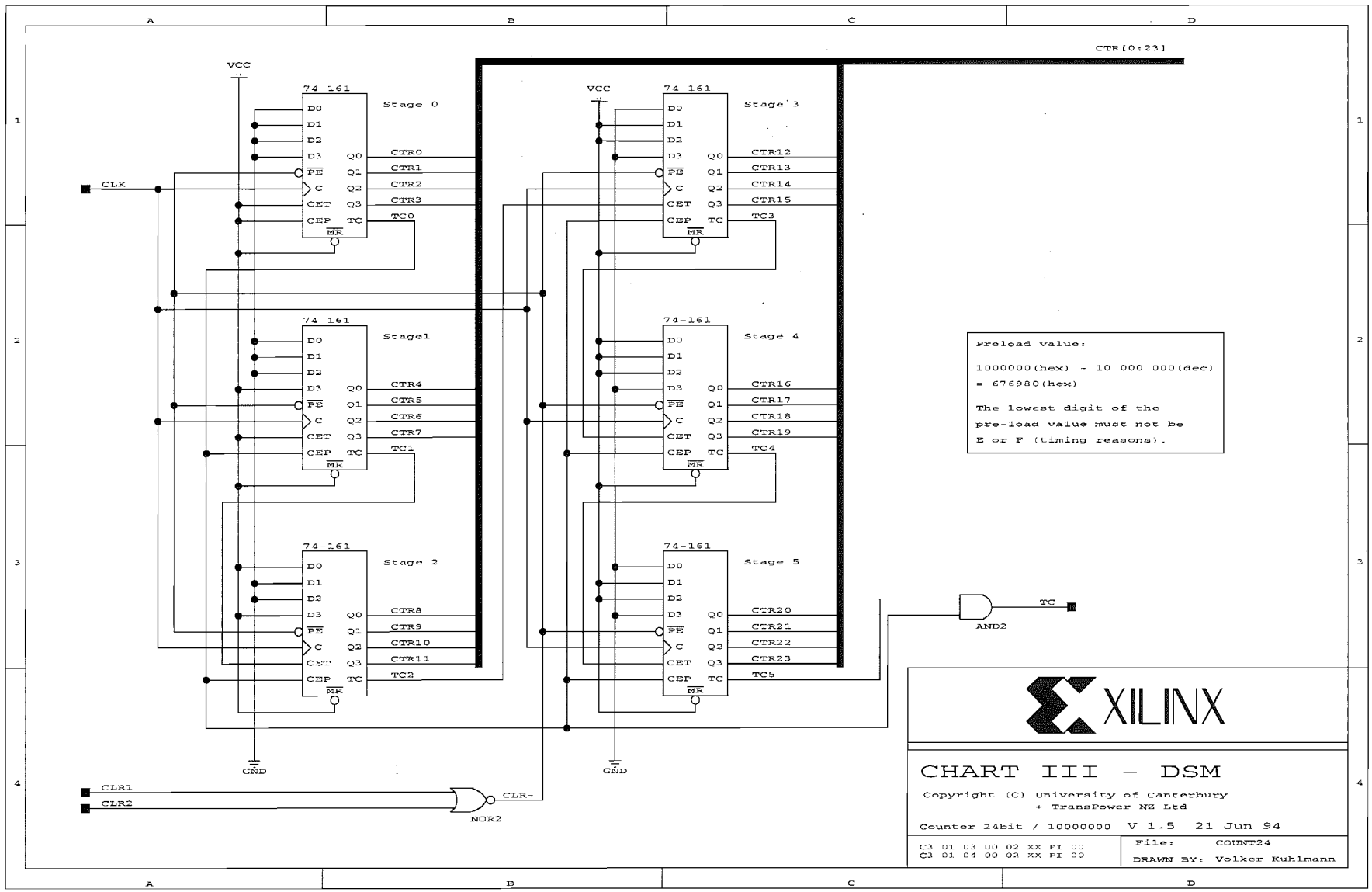
XILINX

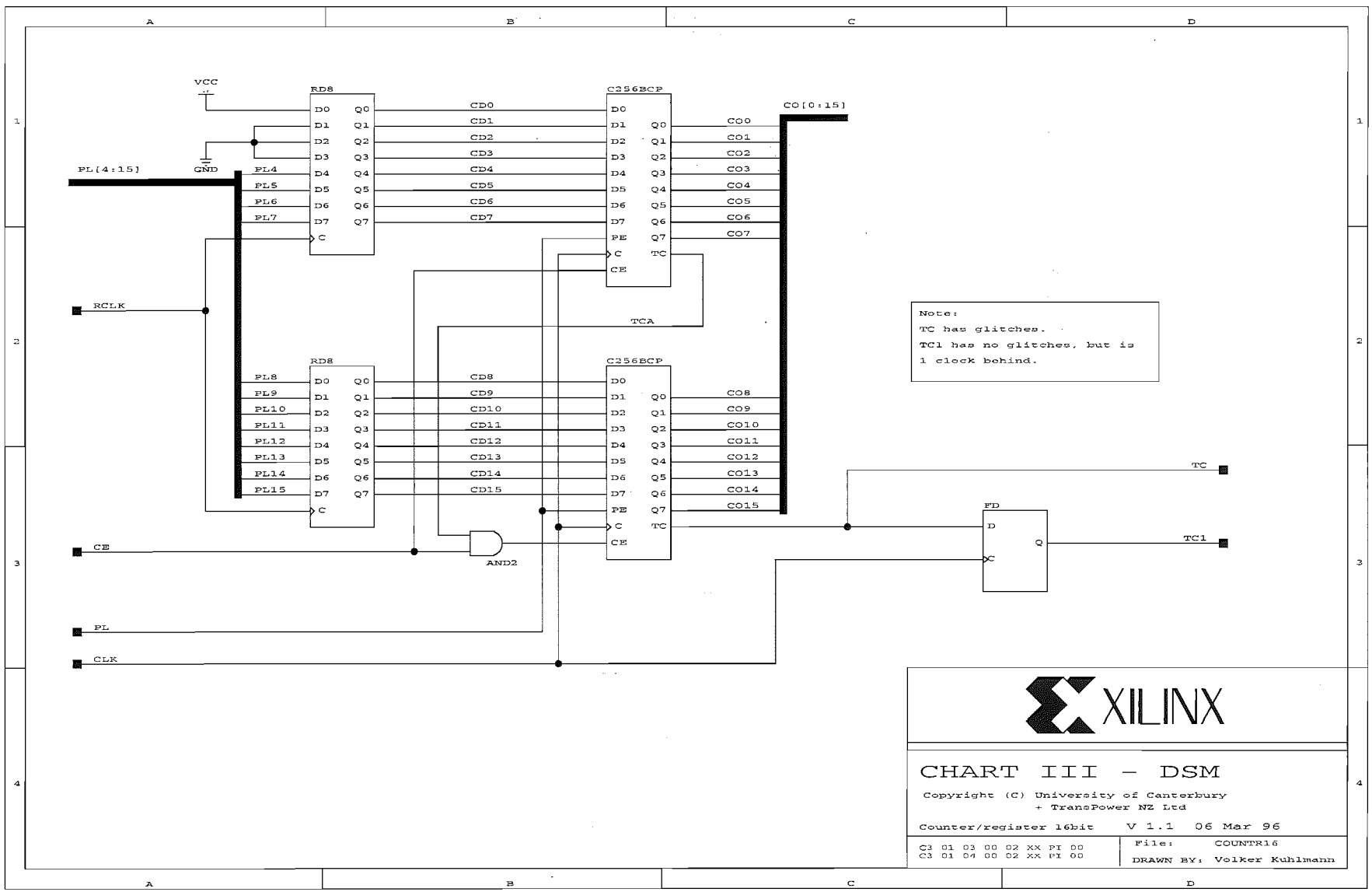
CHART III - DSM

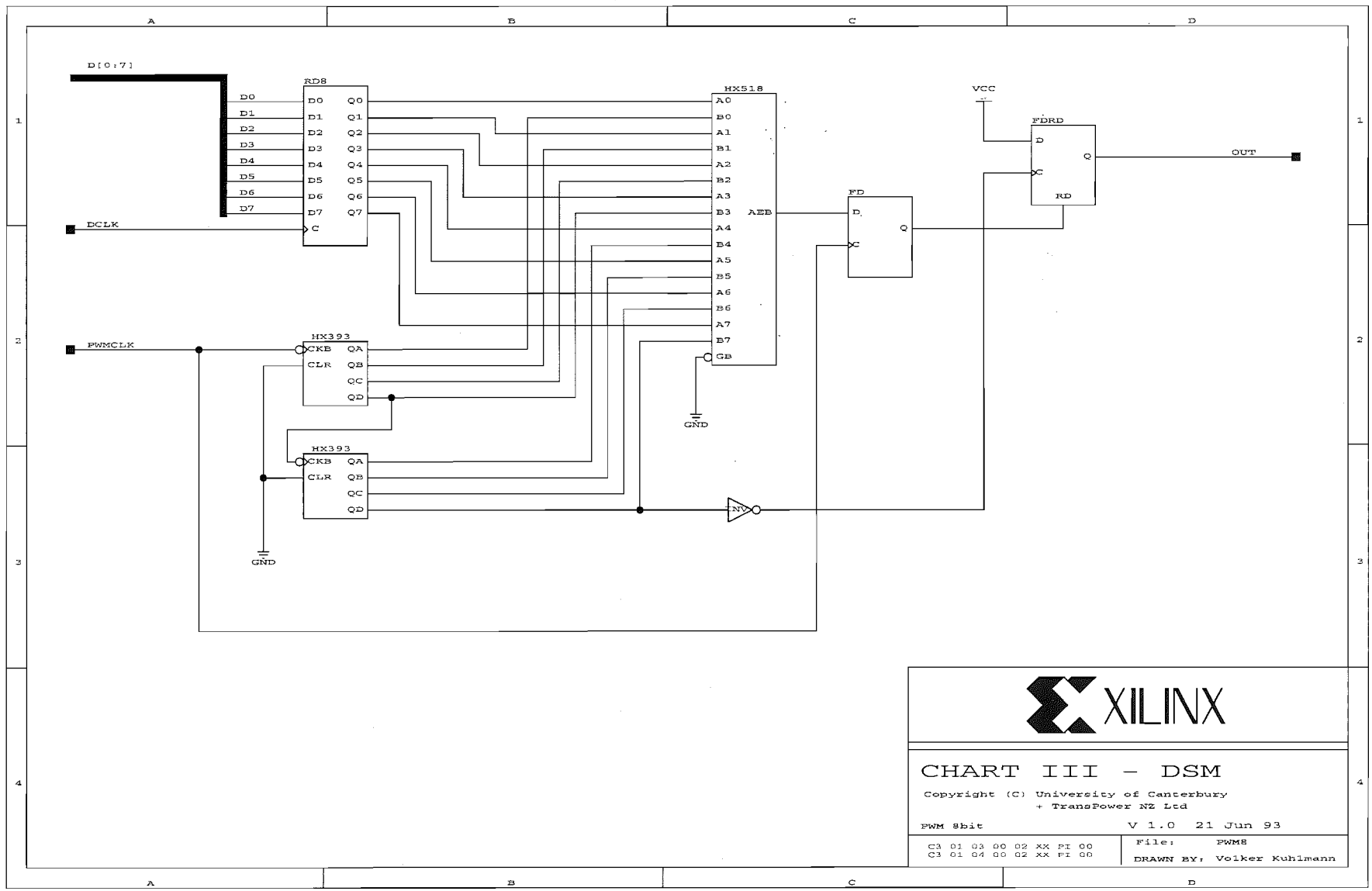
Counter 12bit CLBMAPs

Filename: COUNT12 (from CPSU13)
Sheet: 2 of 3
DRAWN BY: MF-01-08-93, VK 18 Jun 94


A
B
C
D
1
2
3
4
CLK
INV
HX393
CKB
CLR
QA
QB
QC
QD
HX374
D1
D2
D3
D4
D5
D6
D7
D8
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
OEB
CK
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q[0:11]
GND
OE
VCC
DFF
D
Q
CE
C
RD
CAPT
AND2B1
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Counter/Capture 12bit
V 1.3 14 Oct 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: CCAPT12
DRAWN BY: Volker Kuhlmann


CTR[0:23]
VCC
74-161
Stage 0
D0
D1
D2
D3
Q0
PE
Q1
C
Q2
CET
Q3
CEP
TC
MR
CTR0
CTR1
CTR2
CTR3
TC0
CLK
74-161
Stage1
CTR4
CTR5
CTR6
CTR7
TC1
74-161
Stage 2
CTR8
CTR9
CTR10
CTR11
TC2
GND
CLR1
CLR2
CLR-
NOR2
VCC
74-161
Stage 3
CTR12
CTR13
CTR14
CTR15
TC3
74-161
Stage 4
CTR16
CTR17
CTR18
CTR19
TC4
74-161
Stage 5
CTR20
CTR21
CTR22
CTR23
TC5
GND
AND2
TC
Preload value:
1000000(hex) - 10 000 000(dec)
= 676980(hex)
The lowest digit of the
pre-load value must not be
E or F (timing reasons).
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Counter 24bit / 10000000 V 1.5 21 Jun 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: COUNT24
DRAWN BY: Volker Kuhlmann


A
B
C
D
1
2
3
4
VCC
GND
RD8
C256BCP
PL[4:15]
RCLK
CE
PL
CLK
TCA
AND2
FD
TC
TC1
CO[0:15]
Note:
TC has glitches.
TC1 has no glitches, but is
1 clock behind.
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Counter/register 16bit V 1.1 06 Mar 96
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: COUNTR16
DRAWN BY: Volker Kuhlmann



XILINX

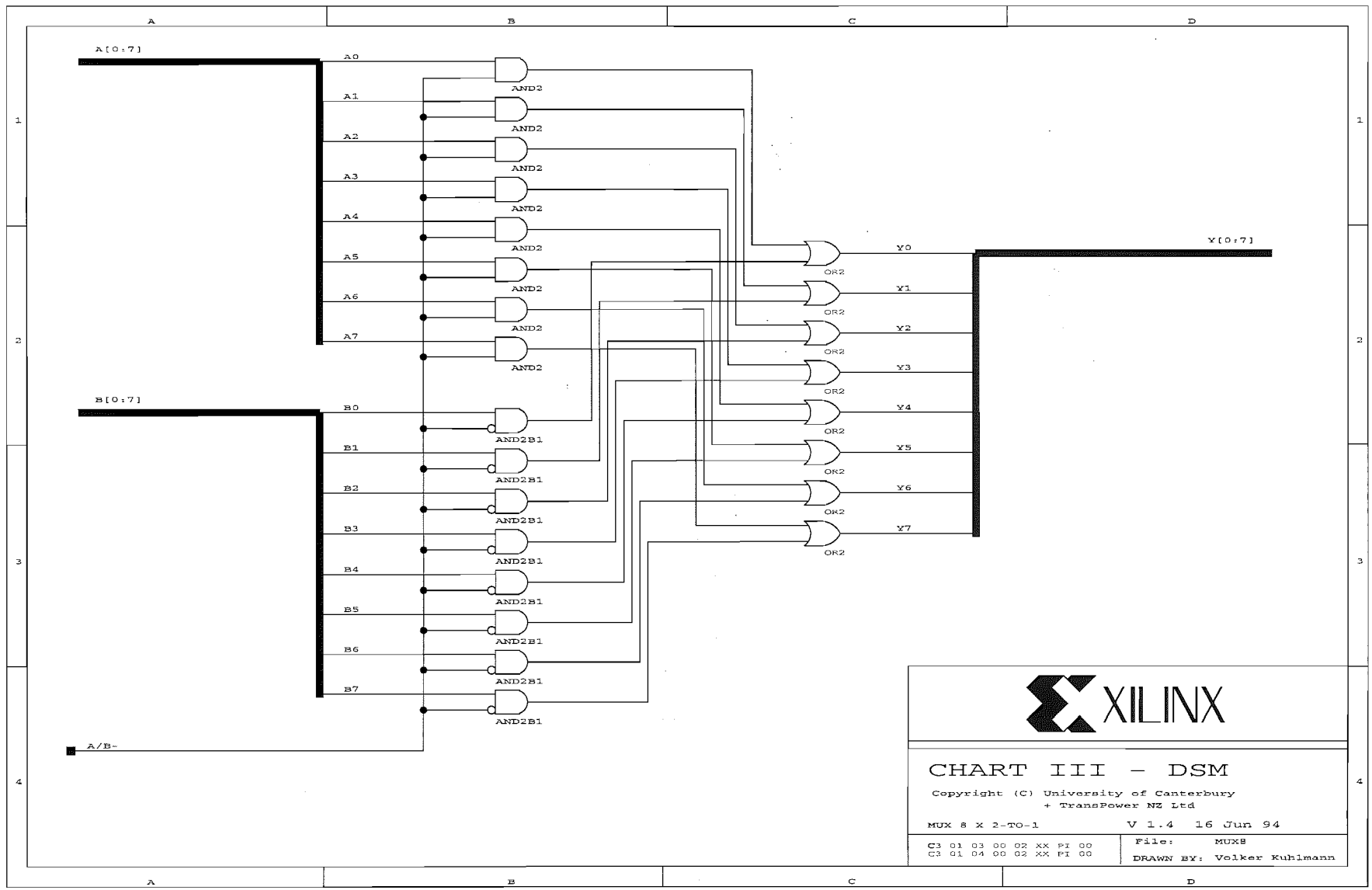
CHART III – DSM

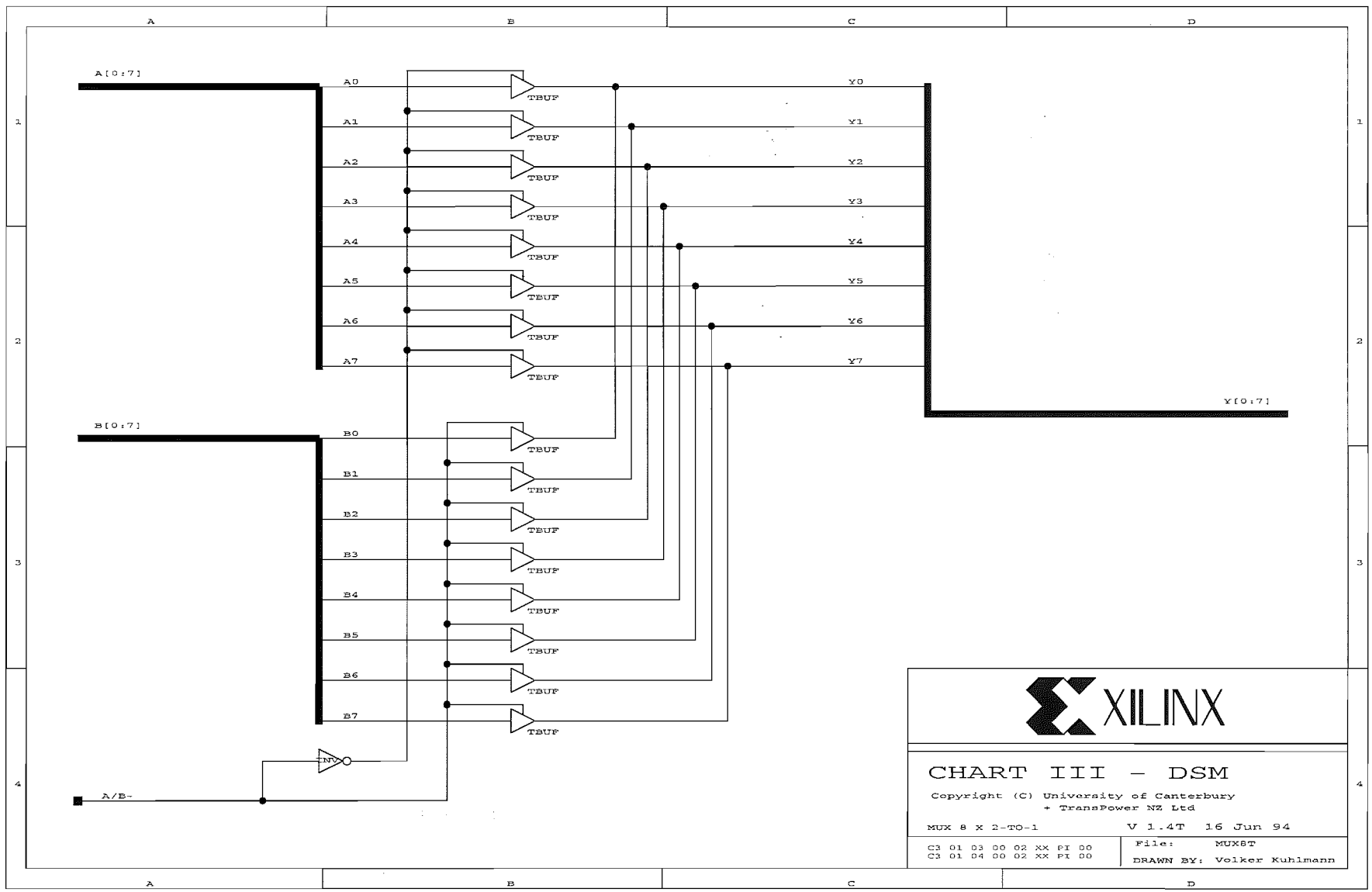
PWM 8bit V 1.0 21 Jun 93

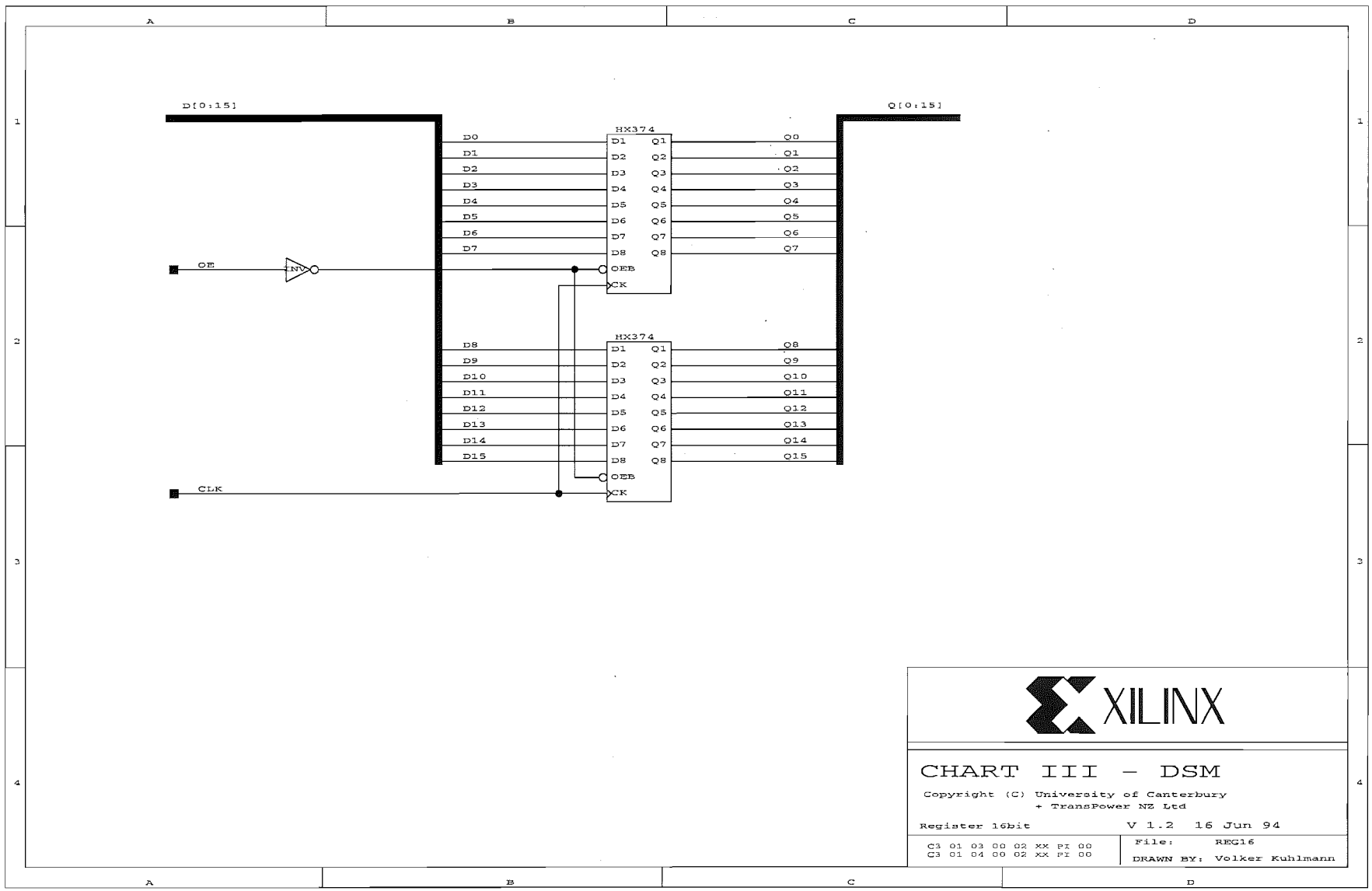
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00

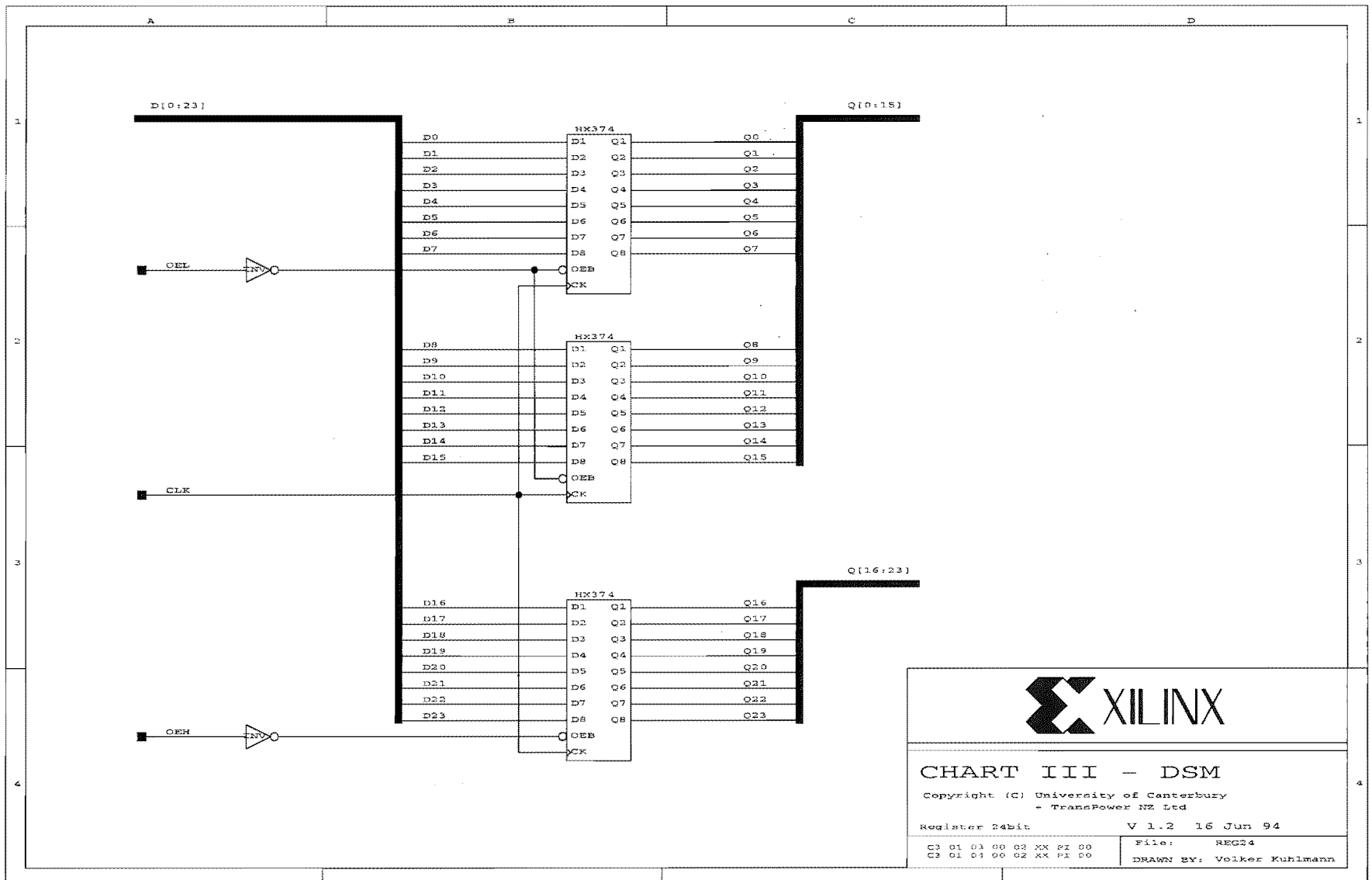
File: PWM8

DRAWN BY: Volker Kuhlmann


A[0:7]
B[0:7]
A/B~
A0
A1
A2
A3
A4
A5
A6
A7
B0
B1
B2
B3
B4
B5
B6
B7
AND2
AND2B1
OR2
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y[0:7]
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
MUX 8 X 2-TO-1
V 1.4 16 Jun 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: MUX8
DRAWN BY: Volker Kuhlmann


A
B
C
D
1
2
3
4
A[0:7]
A0
A1
A2
A3
A4
A5
A6
A7
TBUF
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y[0:7]
B[0:7]
B0
B1
B2
B3
B4
B5
B6
B7
INV
A/B-
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
MUX 8 X 2-TO-1
V 1.4T 16 Jun 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: MUX8T
DRAWN BY: Volker Kuhlmann


D[0:15]
Q[0:15]
HX374
D0
D1
D2
D3
D4
D5
D6
D7
D8
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
OEB
CK
OE
INV
HX374
D8
D9
D10
D11
D12
D13
D14
D15
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
OEB
CK
CLK
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Register 16bit
V 1.2 16 Jun 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: REG16
DRAWN BY: Volker Kuhlmann


D[0:23]
Q[0:15]
Q[16:23]
HX374
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
Q16
Q17
Q18
Q19
Q20
Q21
Q22
Q23
OEB
CK
OEL
CLK
OEH
INV
XILINX
CHART III - DSM
Copyright (C) University of Canterbury
+ TransPower NZ Ltd
Register 24bit
V 1.2 16 Jun 94
C3 01 03 00 02 XX PI 00
C3 01 04 00 02 XX PI 00
File: REG24
DRAWN BY: Volker Kuhlmann

appendix F

FPGA Configuration of the DSM

A configuration for a XILINX FPGA can be downloaded by the development system with the FPGA being in slave serial mode (for other modes refer to table 7.4). Connection to the target system is made through a DIP-8 socket which has the same pinout as a serial PROM used for storing configuration data. If a jumper is provided to switch from slave serial mode to master serial mode, a PROM can be inserted into the socket for mass-production. This approach was chosen for the DSM, the two jumpers concerned are J5 and J6.

To reduce the number of devices which have to be programmed, both FPGAs can be configured in a daisy-chain. Three lines are required for serial configuration. Three jumpers, J7–J9, connect the three lines of each of the two FPGAs for daisy-chain operation. It is also possible to configure both daisy-chained FPGAs in slave serial mode by the DSP, eliminating the need for PROMs completely. This reduces handling costs (programming, etc.) for the PROMs, and allows faster updates. A multiplexer (U28) is provided for switching the 3 lines required for configuring the SRM FPGA between PROM and DSP. The state of the multiplexer is software-controlled.

The development system does not seem to support the generation of a configuration pattern suitable for daisy-chain loading. However, there is enough documentation to do this manually, the procedure is described below.

For merging the configuration patterns of the two FPGAs manually, it is best to output them in the format called "rawbits" by the development system. Files with the rawbits format can be viewed and edited using any text editor, every bit is stored as text "0" or "1" (hex 0x30 and 0x31). Lines not starting with either one of these characters are ignored. Information about how the configuration pattern is composed can be found in [231], chapter 6.

The configuration bit pattern starts with 11111111, and 0010. Following this is a length count, LC. It was not exactly clear how this 24-bit length count should be calculated, but one method states

$$LC = 40 + (140 \times 329) + 4 + (108 \times 285) + 4 = 76888$$

The LC is represented as a 24-bit binary number, and is followed by 1111. The 40 bits described so far make up the header.

The number of bits for the XC3164 configuration is 140×329, followed by 1111. The XC3142A has 108×285 bits, followed by 1111.

The bit stream is completed with a number of pad bits, which all have to be "1". It was observed that the FPGAs do not start to operate before receiving a number of pad bits. The exact number could not be determined, but having a few more than necessary has no negative effects. In this case 4 were sufficient.

The structure of the combined rawbits data is reproduced below:

```
;new header for daisy chain:
11111111
0010
; LC = 40 + (140*329) + 4 + (108*285) + 4 = 76888
00000001001011000011011000
1111

;data for 3164  (140*329 bits):
00110111111111111111111111111111111111111111111111111111111111111...
01111110111011101111111011111110111101101111111011101110111101101...
....
01111111111111111111111111111111111111111111111111111111111111111...

;tail bits:
1111

;data bits for 3142A  (108*285 bits):
00111111111111111111111111111111111111111111111111101111111101...
01110110111111101111111011111110111011101100111011001111011001111...
...
01111111111111111111111111111111111111111111111111111111111111111...

;tail bits:
1111

;pad bits:
1111
```

A C program converts the bits of the combined configuration patterns into an initialised C variable, which can then be used with a C compiler and the DSM firmware:

```
#include <stdio.h>
#include <string.h>

#ifdef BORLAND_C
  #include <process.h>
#endif


void main(int argc, char *argv[])
{
FILE *infile;
char buffer[256], *temp;
int len;
unsigned long numwords, numbits;
unsigned long data, ptrbit;

    if (argc != 2)
        {
        fprintf (stderr, "Requires input file name.\n\n");
        _exit(1);
        }

    infile = fopen (argv[1], "rt");
    if (infile == NULL)
        {
        perror (argv[1]);
        _exit (1);
        }

    data = 0L;
    ptrbit = 1L;
    numwords = 0;
    numbits = 0;
    printf ("/* File: %s */\n", argv[1]);
    printf ("unsigned long Xconfig[] = {\n");
    while (fgets (buffer, 256, infile) != NULL)
        {
        if (buffer[0] != '0' && buffer[0] != '1') continue;
```

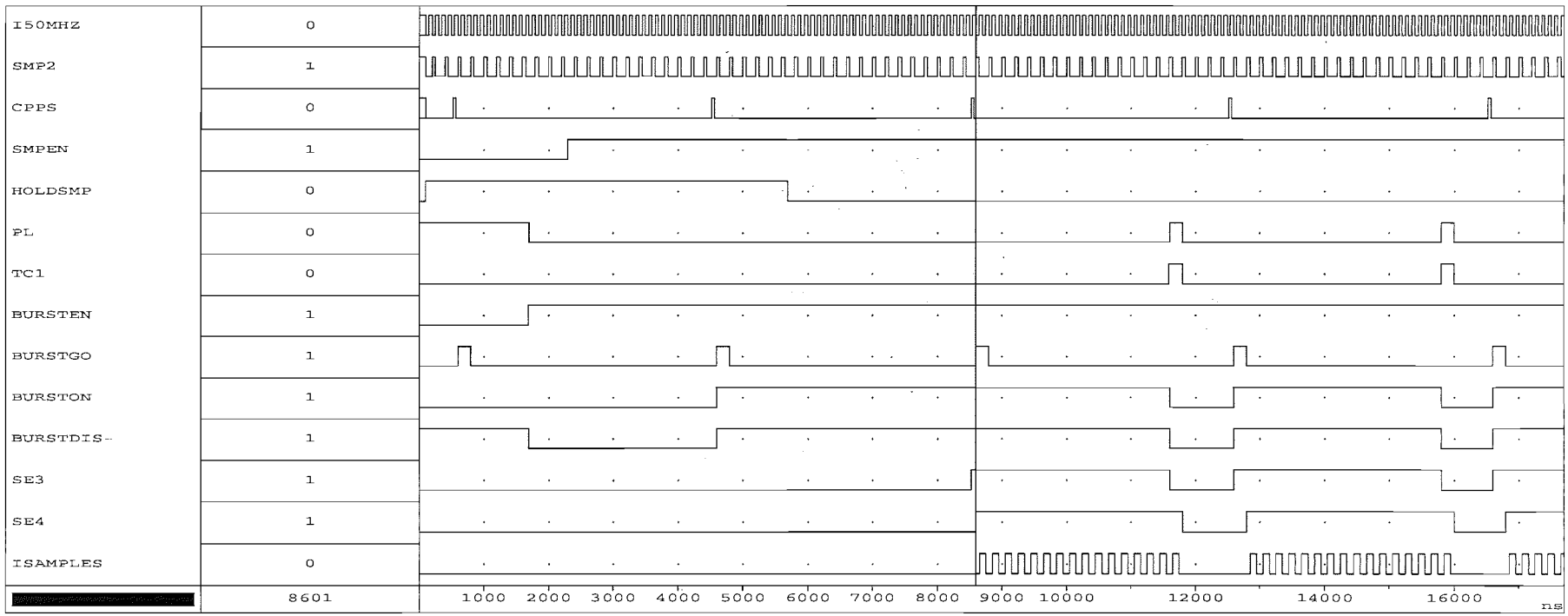
```
        len = strlen(buffer)-1;
        buffer[len] = 0;
        for (temp = buffer; ; temp++)
            {
            if (*temp == 0) break;
            if (*temp == '1') data |= ptrbit;
            numbits++;

            ptrbit <<= 1;
            if (ptrbit == 0)
                {
                ptrbit = 1L;
                printf ("0x%08lx,\n", data);
                numwords++;
                data = 0L;
                }
            } /* end for */
        } /* end while */
    printf ("0x%08lx};\n",data);
    numwords++;
    printf ("unsigned long Xconfig_size_words = %lu;\n", numwords);
    printf ("unsigned long Xconfig_size_bits = %lu;\n", numbits);
}
```

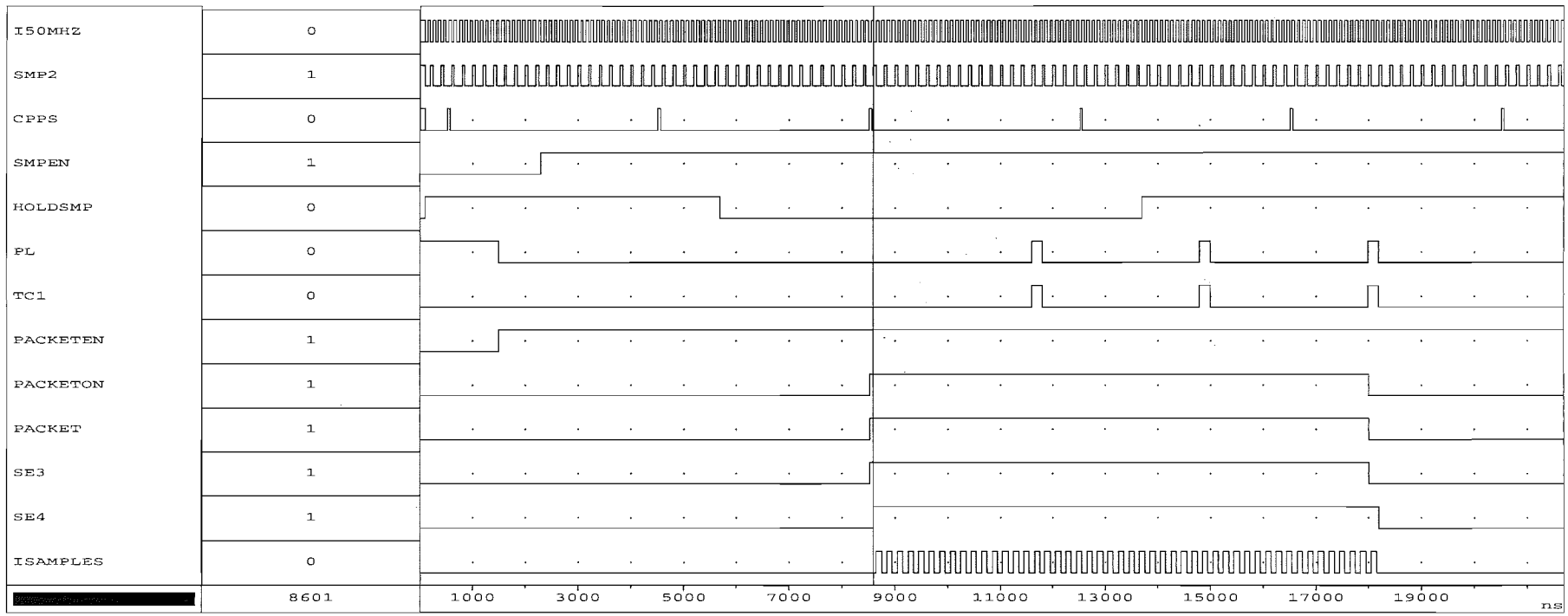
appendix G

Timing Diagrams of the Sampling Clock

These two timing diagrams show the details of the DSM's FPGA logic after the changes described in chapter 8.



Timing diagram for burst sampling mode. SMP2 is the internal sampling clock, which is gated onto the TSB (time stamping bus) by SE4. CPPS is a once-per-second pulse which is generated out of 1pps and an internal 1 s clock combined. To enable sampling, SMPEN must be 1 and HOLDSMP 0. For burst sampling, BURSTEN must be 1. ISAMPLES is the signal appearing on the TSB.
It can be seen that at the beginning of each second (CPPS), a burst of 16 samples is generated (ISAMPLES).



Timing diagram for packet sampling mode. SMP2 is the internal sampling clock, which is gated onto the TSB (time stamping bus) by SE4. CPPS is a once-per-second pulse which is generated out of 1pps and an internal 1 s clock combined. To enable sampling, SMPEN must be 1 and HOLDSMP 0. For packet sampling, PACKETEN must be 1. ISAMPLES is the signal appearing on the TSB. It can be seen that at the beginning of the second (CPPS) following HOLDSMP being set to 1, sampling continues (ISAMPLES) until the current packet has been completed. The packet size is 16 samples, and 3 packets are completed (there are 48 samples total).

appendix H

Photographs

H.1 CHART System

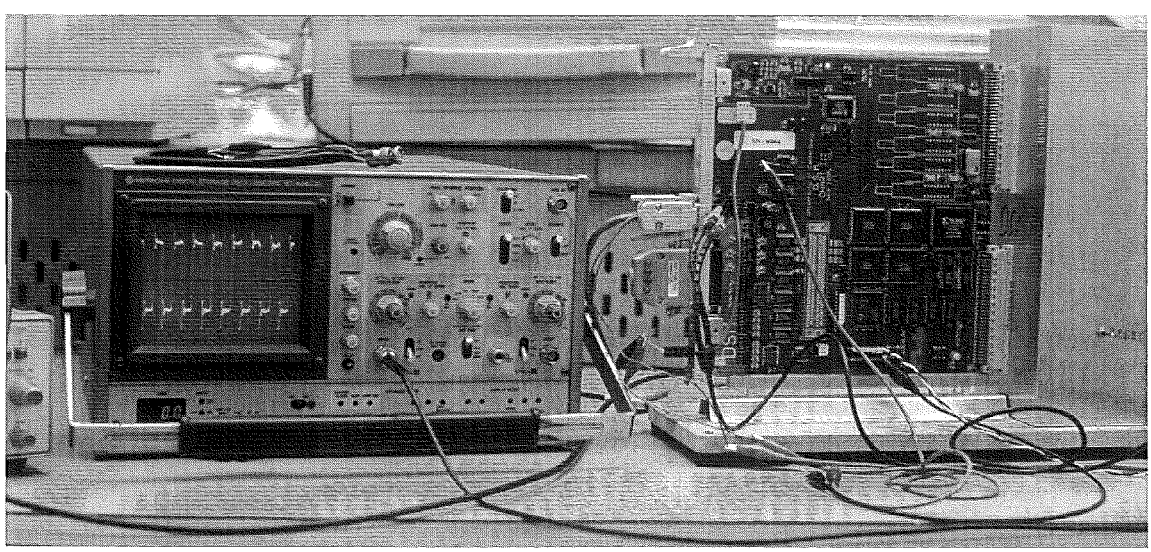


Photo 2: The DSM board in the test rack, with attached test equipment.

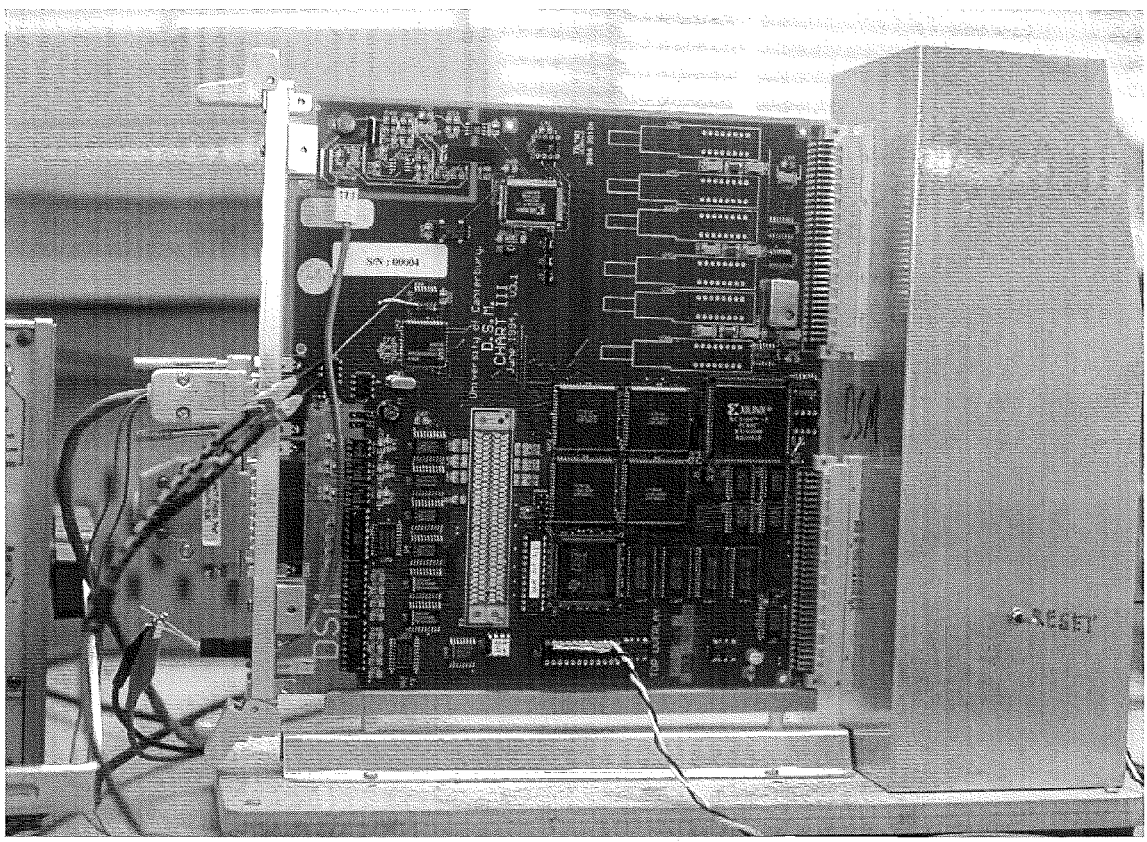


Photo 3: The DSM board in the test rack. In the upper right corner is the space for the F/O devices. The serial test cable is the one with the DB9 connection approx. in the middle of the front panel. Below the test cable connector is the PARIO test adaptor with 4 LEDs in a DB25 case, behind 2 other leads. The mains reference connector and the front panel LED are at the bottom of the front panel. The MIX address decoder GAL (DIP24-300 case) at the bottom of the PCB is replaced with a resistor network. The reset switch is connected to this network as well.

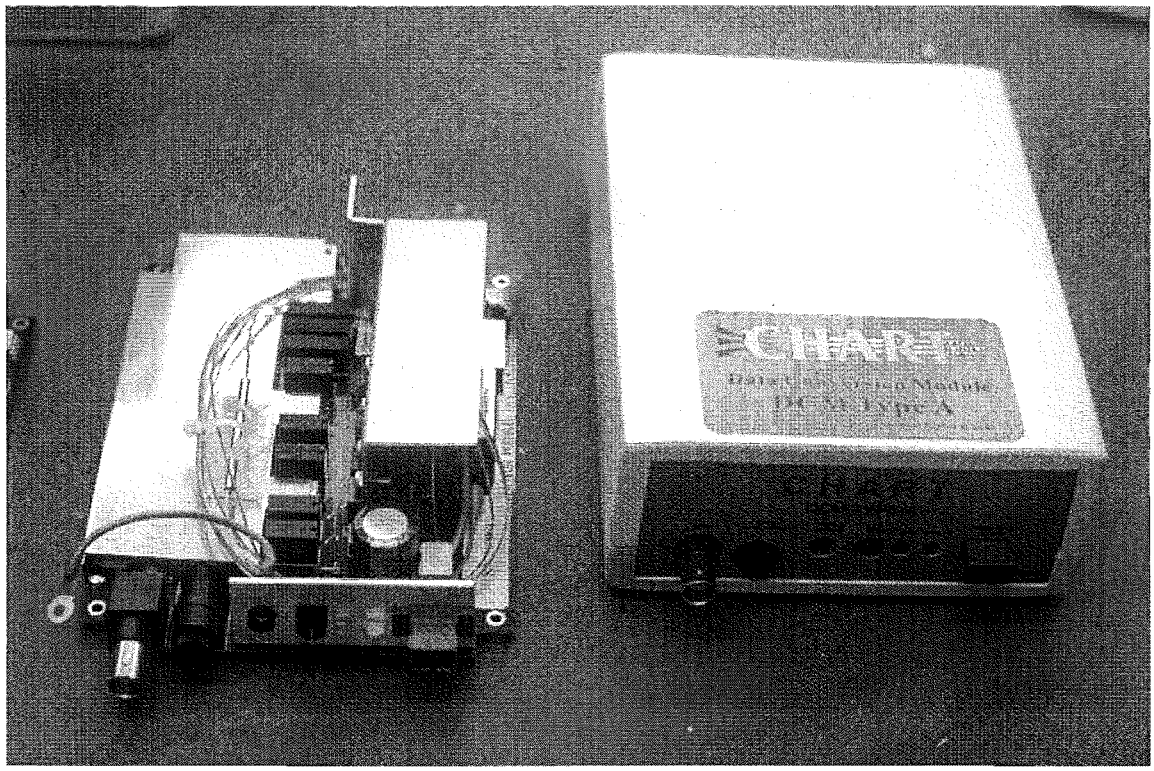


Photo 4: The DCM which contains ADC and F/O converter, with and without case.

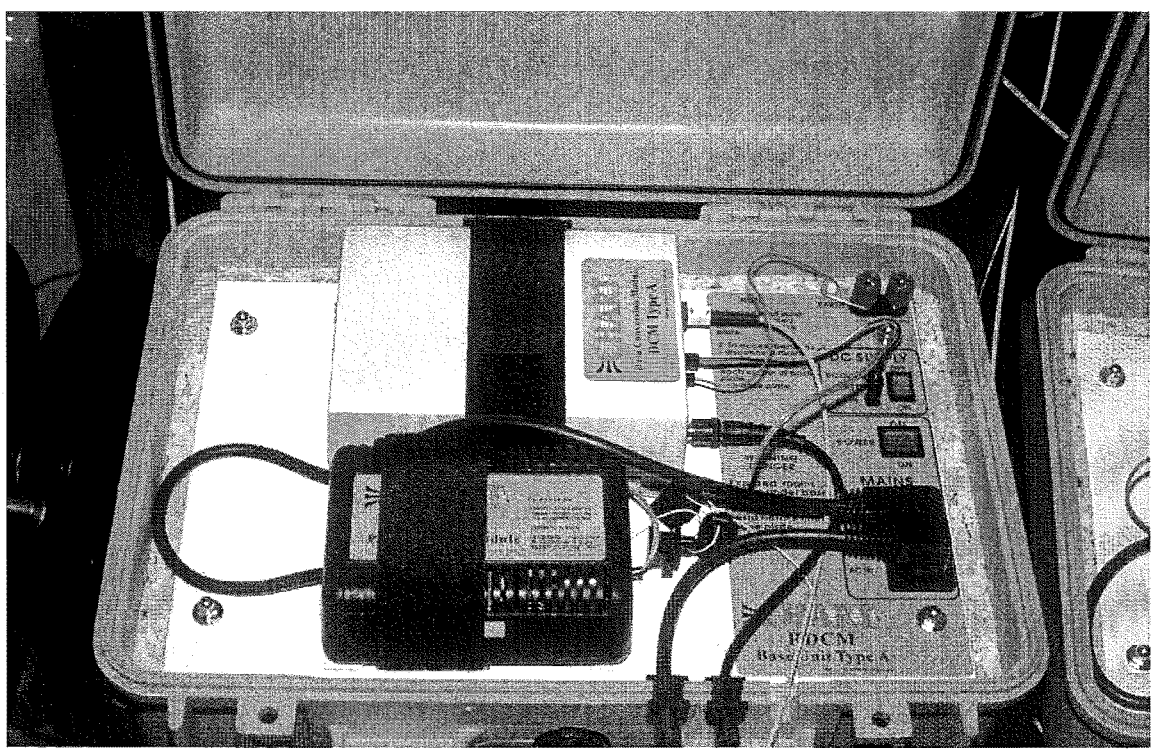


Photo 5: The RDCM in a ruggedised, weatherproof case. Batteries are located in the bottom of the case.

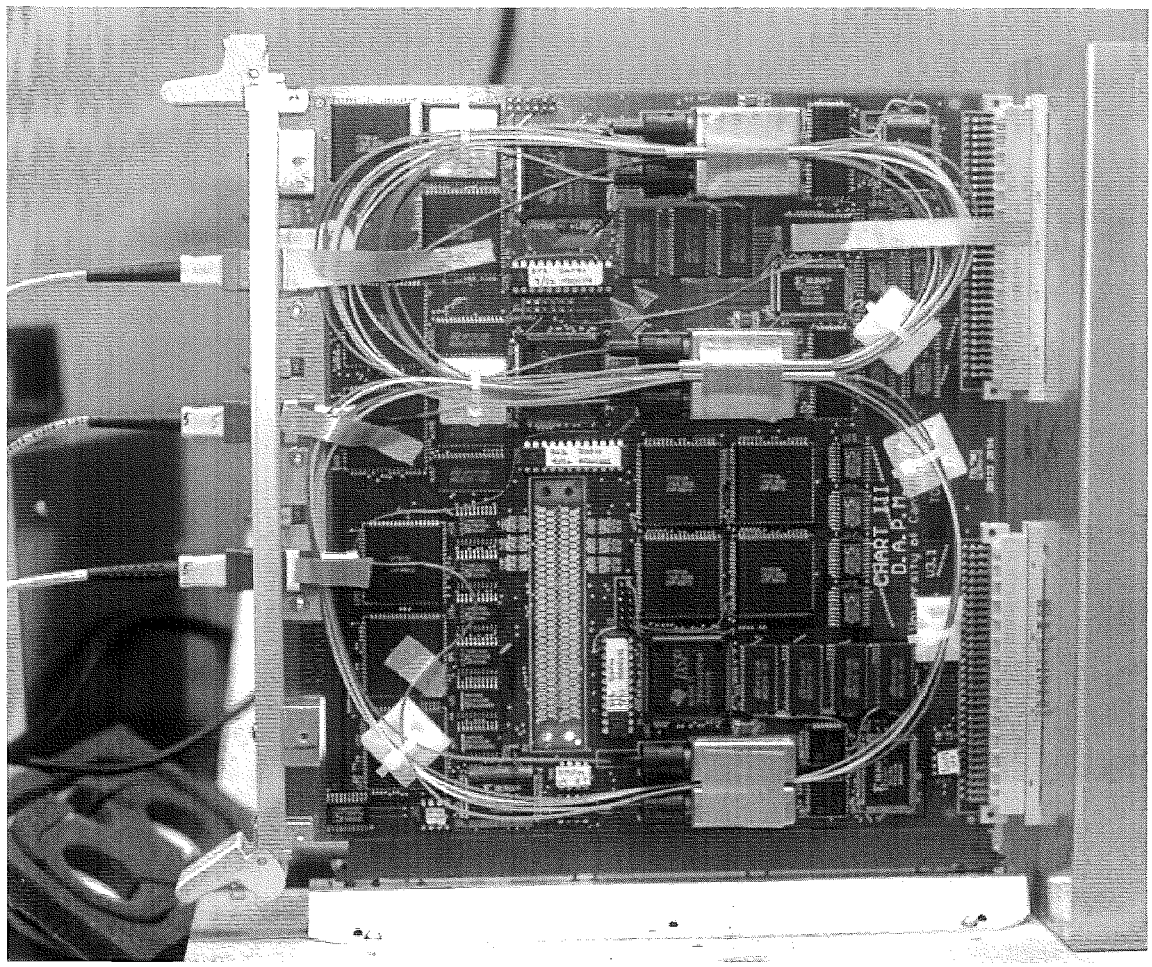


Photo 6: The DAPM board, with 3 F/O cables attached

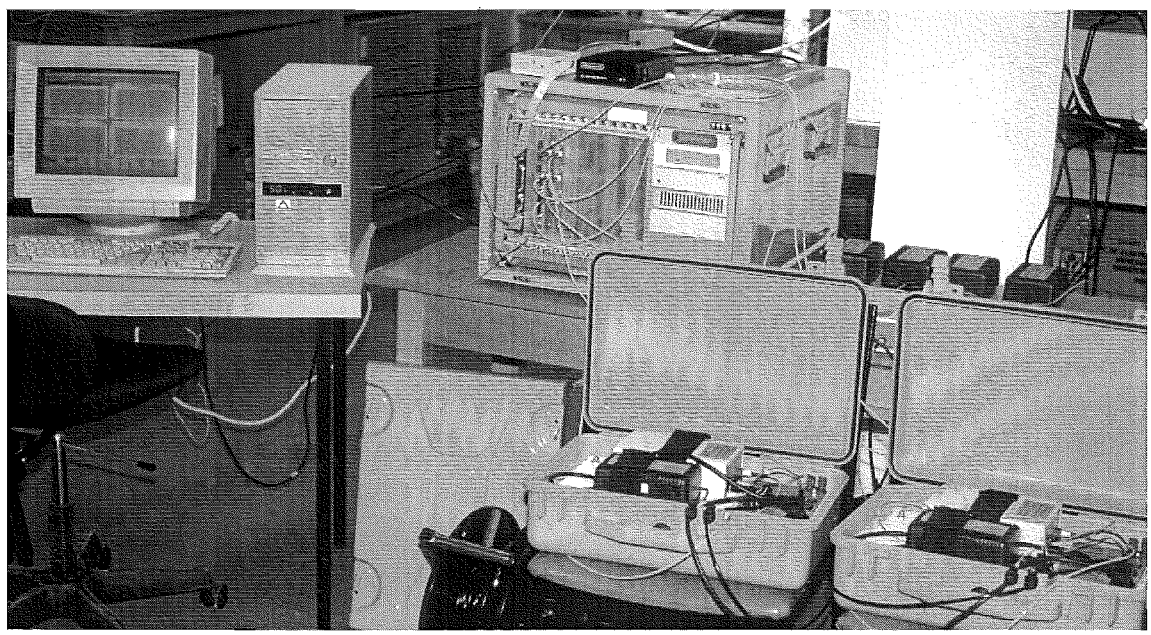


Photo 7: A complete CHART unit. The CADU is on the left, the RDCMS are in the foreground, and the PPU is standing on the table behind. The GPS receiver is visible on top of the PPU.

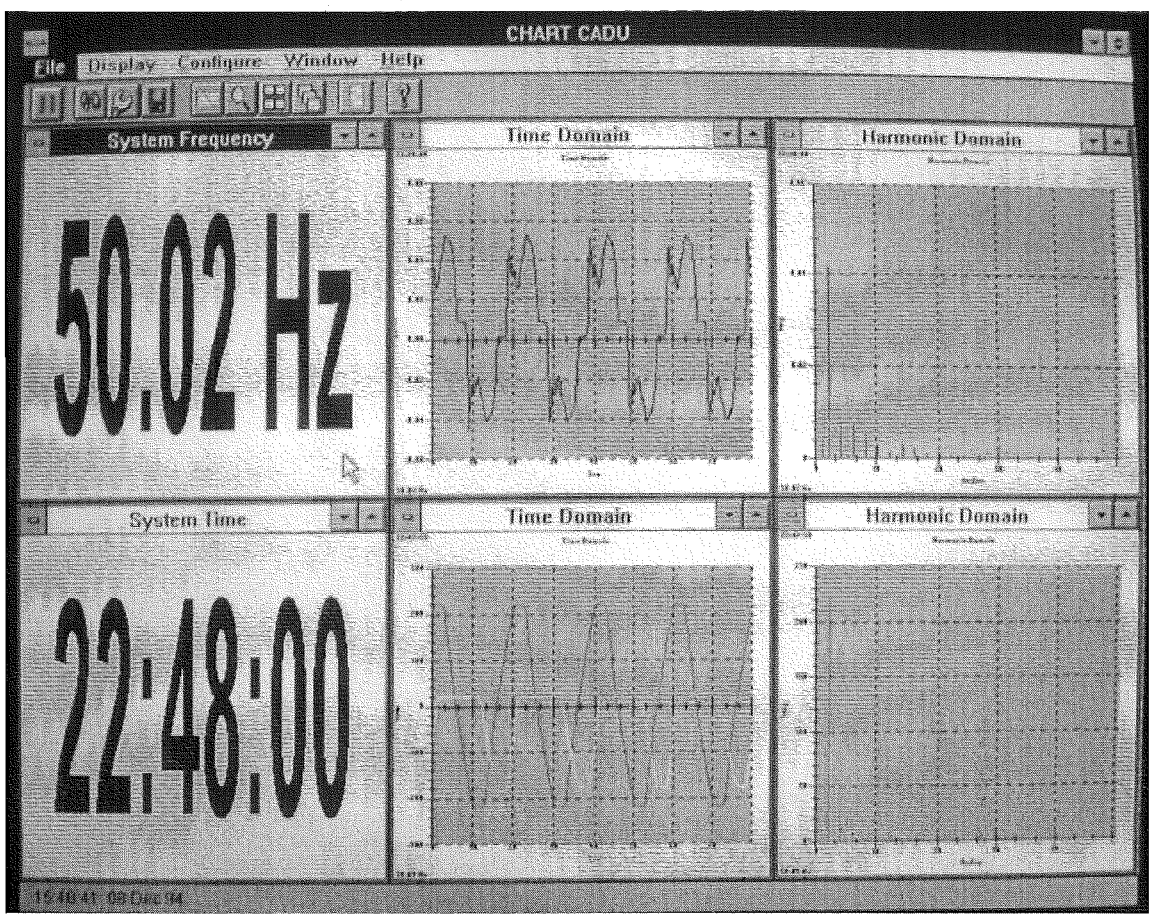


Photo 8: A CADU work screen, showing the measured mains frequency, system time, and voltages and currents of a 3-phase system in the time and frequency domain.

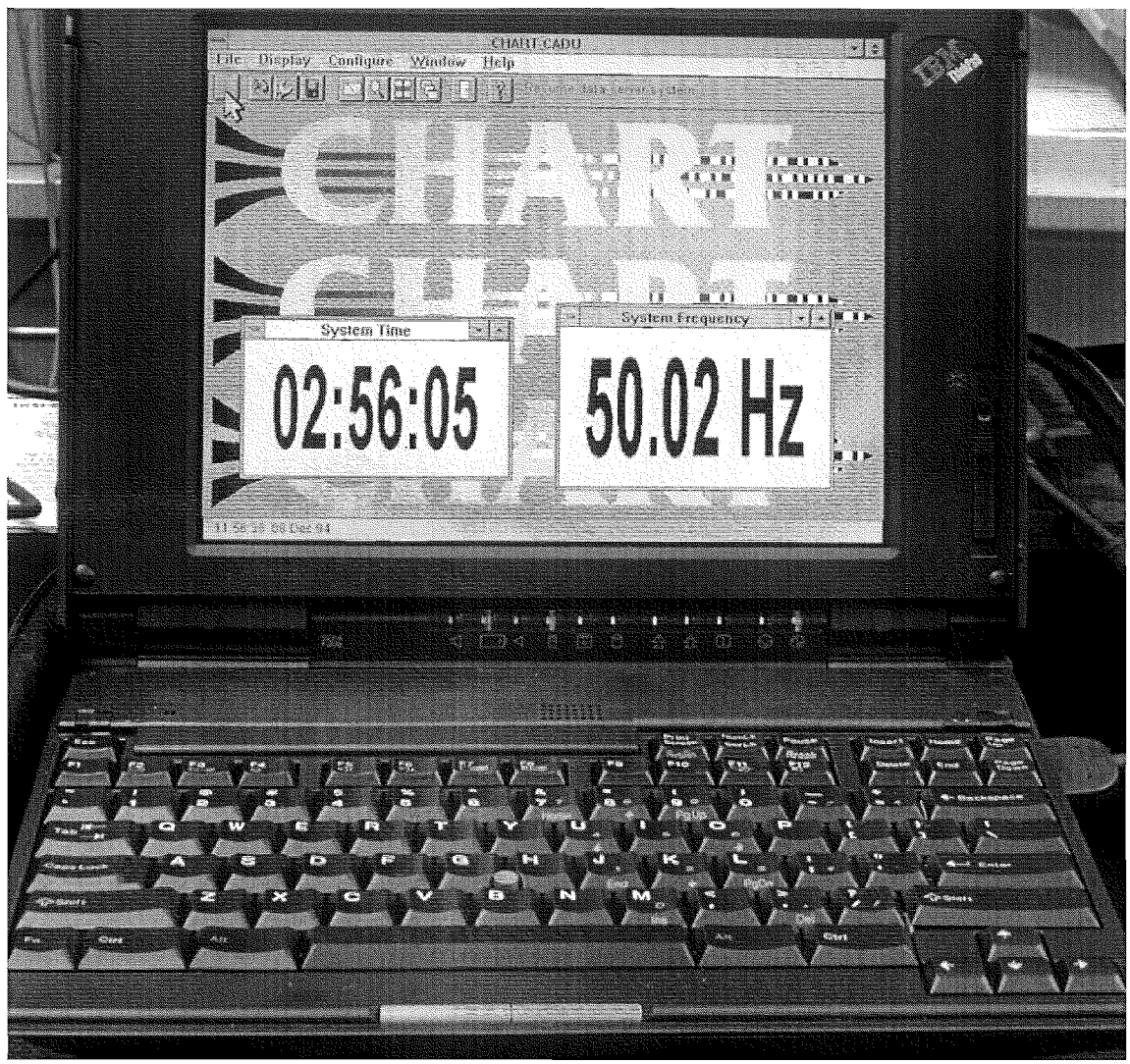


Photo 9: This laptop computer can be used as CADU. The display shows the measured mains frequency and the system time, as given by the DSM.

H.2 Ripple Injection Field Tests

For a discussion of the ripple injection field tests refer to section 9.1.

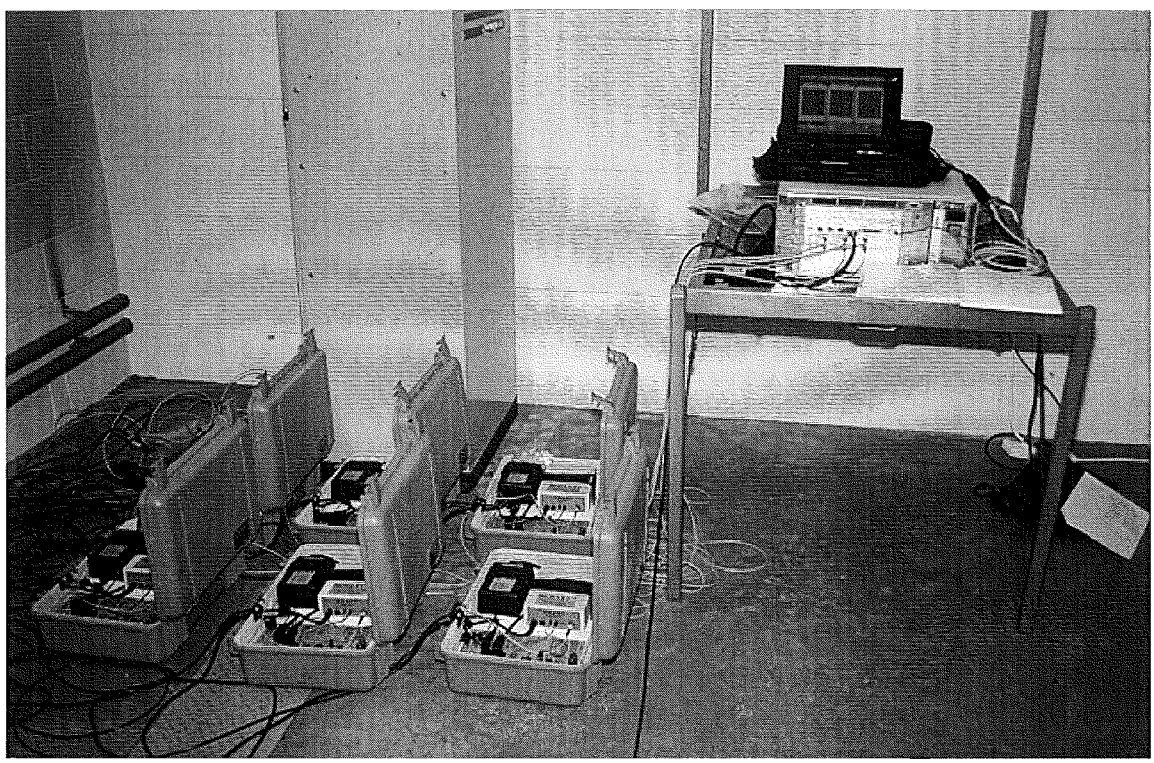


Photo 10: CHART set up at the Papanui substation. The 6 boxes on the left are the RDCMS, the PPU is standing on the table. The laptop on top of the PPU is used by the operator to control PPU functions.

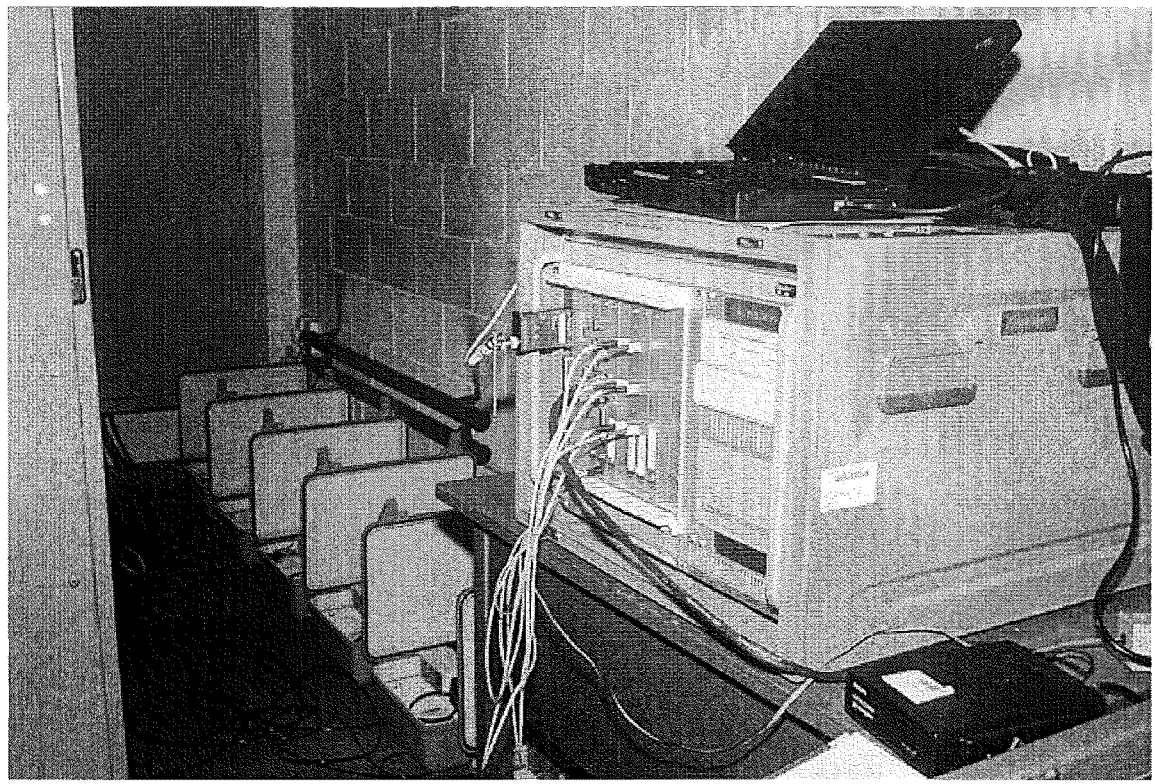


Photo 11: CHART set up at the Pages substation. The RDCMs are in a line along the wall, the PPU sits on the table, with the laptop on top. The GPS receiver is the black box in the lower right corner of the photo.

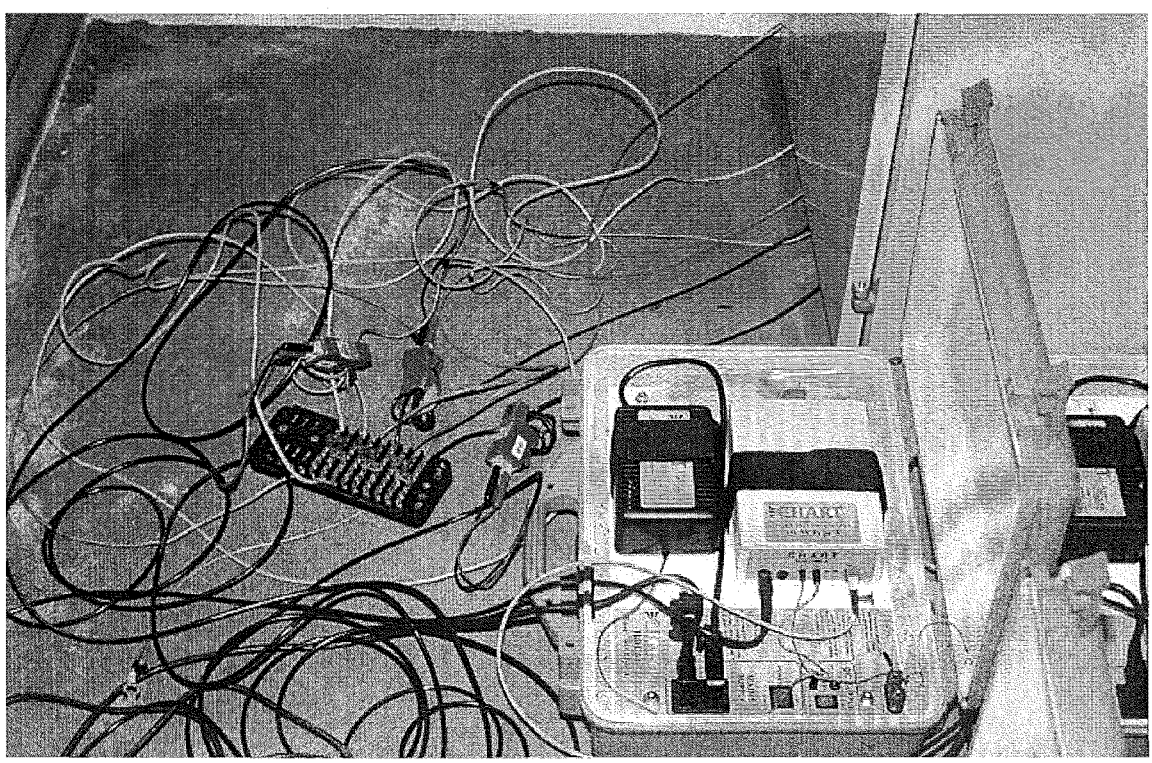


Photo 12: An RDCM connected to the substation circuitry. The 3 current clips are clearly visible. Papanui substation.

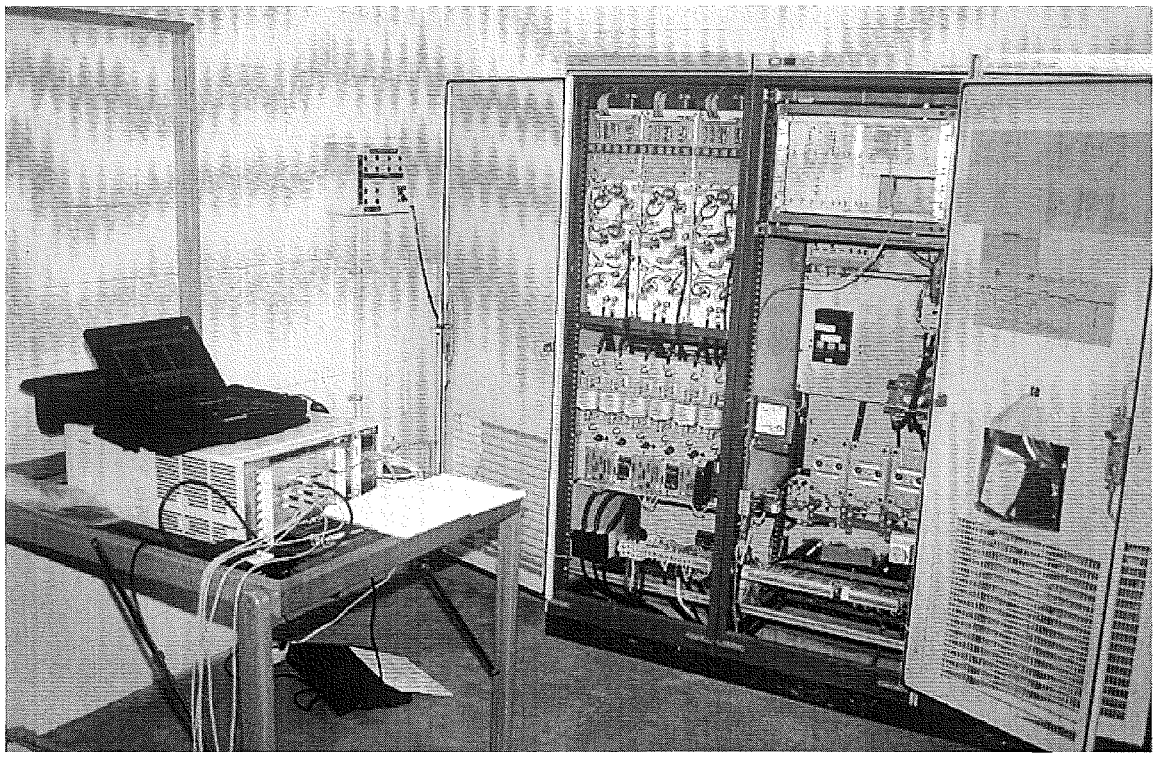


Photo 13: The ripple injection equipment at Papanui substation. All 3 ripple injectors in Christchurch are of the same type. This shows the converter, inverter and the control circuitry.

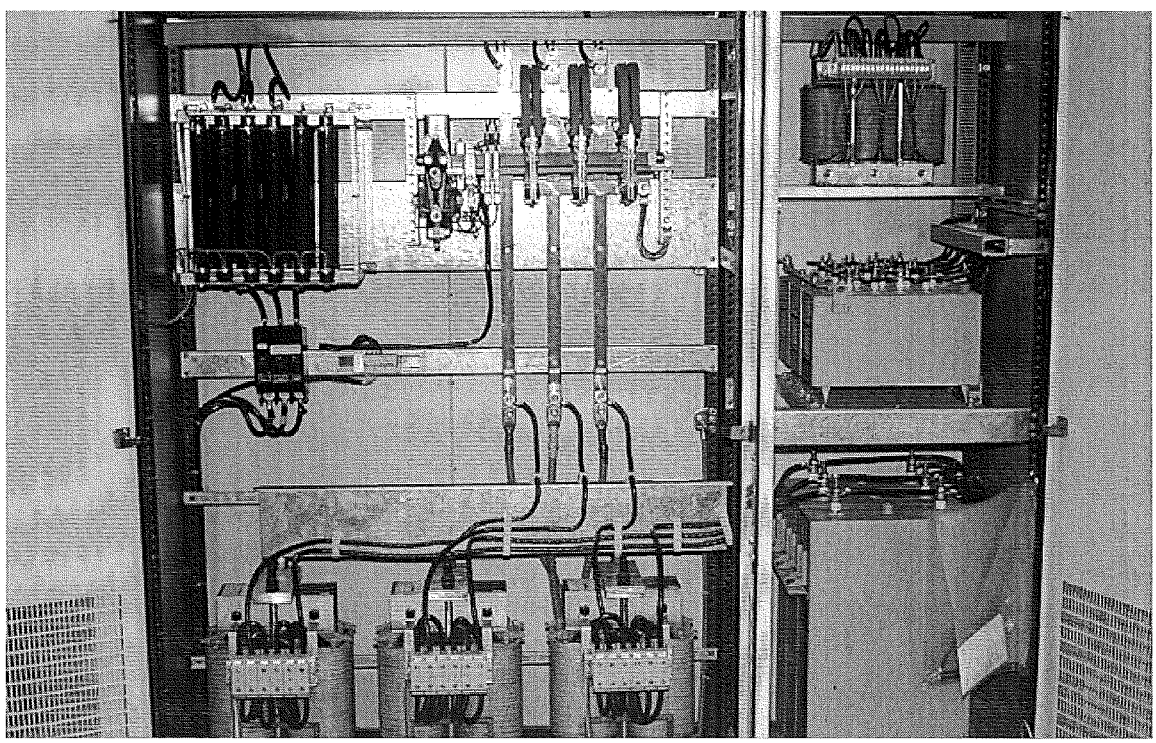


Photo 14: The ripple injection equipment at Papanui substation. This shows the 50 Hz and 250 Hz inductors and capacitors.

References

[1] H. Abdel-Ghaffar. Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays. *IEEE Transactions on Communications*, 50(10): 1652–1661, 2002. ISSN 0090-6778.

[2] K. Abdel-Rahman, L. Mill, A. Phadke, J. D. L. Ree, and Y. Liu. Internet based wide area information sharing and its roles in power system state estimation. In *IEEE Power Engineering Society Winter Meeting*, volume 2, pages 470–475, 2001.

[3] Acoustics research group. URL `http://www.elec.canterbury.ac.nz/research/acoustics/`. Linux driver for PLX PCI9054 available on request; contact Michael Hayes `m.hayes@elec.canterbury.ac.nz`, University of Canterbury, Electrical Engineering, Christchurch, New Zealand.

[4] Advanced Micro Devices, Inc., 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088-3453, USA. *Am7968/A7969* TAXI*chip Data Sheet and Technical Manual*, 1992. BAN-14.5M-9/92-0, 17240A.

[5] M. Aida, T. Sato, and C. Miki. Dissemination of time signal and standard frequency. dissemination by means of tv signals, LORAN-C and GPS. *Journal of the Communications Research Laboratory*, 39(1):161–167, Mar. 1992. ISSN 0914-9260.

[6] M. Ando. *Fault Location Techniques for HVDC Lines: Analysis, Development, Simulation and Field-Data Evaluation.* PhD thesis, Washington State University, 1984.

[7] M. Ando, E. O. Schweitzer, and R. A. Baker. Development of field data evaluation of single-end fault locator for two-terminal HVDC transmission lines. *IEEE Transactions*, PAS-104(12): 3524–3537, 1985.

[8] M. Ando, E. O. Schweitzer, and R. A. Baker. Fault location techniques for high-voltage DC lines. Final Report EPRI Report No. EL-4331, Washington State University, Dec. 1985. Research Project 2150-1.

[9] AT&T Microelectronics, Solid State Technology Center, 9999 Hamilton Blvd., Breinigsville, PA 18031-9359, USA. *ODL50 Series II Lightwave Data Link*, Jan. 1994. Document DS93-120LWP.

[10] AT&T Microelectronics, Solid State Technology Center, 9999 Hamilton Blvd., Breinigsville, PA 18031-9359, USA. *ODL70 Series II Lightwave Data Link*, Jan. 1994. Document DS93-121LWP.

[11] J. Axford. CHART *Technical Reference.* University of Canterbury, Christchurch, New Zealand, Oct. 1994. Version 1.00.

[12] J. Axford. CHART *User Manual.* University of Canterbury, Christchurch, New Zealand, Oct. 1994. Version 1.00.

[13] S. Baird, J. Coughlan, R. Halsall, J. Hartley, W. Haynes, and T. Parthipan. A PMC based ADC card for CMS tracker readout. 1999? URL `http://lebwshop.home.cern.ch/lebwshop/LEB99_Book/DAQ/coughlan.pdf`.

[14] F. Basegmez. Extending a scientific application with scripting capabilities. *IEEE Computing in Science and Engineering*, 4(6):52–59, 2002. ISSN 1521-9615. [see also IEEE Computational Science and Engineering].

[15] A. Batista, A. Combo, J. Sousa, and C. Varandas. A distributed, hardware reconfigurable and packet switched real-time control and data acquisition system. *Fusion Engineering and Design*, 60(3):443–448, 2002. ISSN 0920-3796.

[16] D. M. Beazley and P. S. Lomdahl. Extensible message passing application development and debugging with Python. In *Proceedings of the 11th International Parallel Processing Symposium*, pages 650–655, 1997.

[17] C. Beetham and J.-B. Ribes. GPS precision timing at CERN. In *Proceedings of the Particle Accelerator Conference*, volume 2, pages 765–767, 1999.

[18] K. M. Borkowski. The Persian calendar for 3000 years. *Earth, Moon, and Planets: an International Journal of Comparative Planetology*, 74(3):223–230, 1997. ISSN 0167-9295.

[19] R. N. Bracewell. *The Fourier Transform & Its Applications.* McGraw-Hill, 2nd edition, 1978. ISBN 0-07-007013-X.

[20] R. N. Bracewell. *The Fourier Transform & Its Applications.* McGraw-Hill, 3rd edition, June 1999. ISBN 0-07-303938-1.

[21] S. M. Brahma and A. A. Girgis. Fault location on a transmission line using synchronized voltage measurements. *IEEE Transactions on Power Delivery*, 2004. ISSN 0885-8977. Accepted for future publication.

[22] S. Bronson. *VPN PPP-SSH Mini-HOWTO.* Linux Documentation Project, 29 July 2001. Version 1.00.

[23] M. Caporaloni and R. Ambrosini. Time synchronization measurements with a combination of a GPS receiver and the internet. *European Journal of Physics*, 22(4):351–359, July 2001. ISSN 0143-0807. URL `http://stacks.iop.org/0143-0807/22/351`.

[24] A. B. Carlson. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication.* McGraw-Hill, New York, 3rd edition, 1986. ISBN 0-07-100560-9.

[25] S. Chen. *Analysis of HVDC Convertor Transformer Core Saturation Instability, and Design of a Data Acquisition System for its Assessment.* PhD thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Jan. 1996.

[26] S. Chen. A quantitative analysis of the data acquisition requirements for measuring power quality phenomena. *IEEE Transactions on Power Delivery*, 18(4):1575–1577, Oct. 2003. ISSN 0885-8977.

[27] S. Chen. Open design of networked power quality monitoring systems. *IEEE Transactions on Instrumentation and Measurement*, 53(2):597–601, 2004. ISSN 0018-9456.

[28] S. Chen and M. B. Dewe. System and hardware considerations of a flexible multi-channel real time data acquisition system for power quality assessment. In Proceedings of the 7th International Conference on Harmonics and Quality of Power (ICHQP) [179], pages 189–195. ISBN none.

[29] S. Chen and M. B. Dewe. Virtual operating system design in a flexible multi-channel real time data acquisition system for power quality assessment. In Proceedings of the 7th International Conference on Harmonics and Quality of Power (ICHQP) [179], pages 182–188. ISBN none.

[30] S. Chen and A. J. V. Miller. *RDCM Hardware Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[31] S. Chen and X. L. Wang. Embracing advance information technology in power quality monitoring. In *IEEE Power Engineering Society Winter Meeting*, volume 2, pages 1466–1471. The Institute of Electrical and Electronics Engineers, Inc., 27–31 Jan. 2002.

[32] S. Chen, C. L. Zhang, and Y. Z. Liu. A multi-channel monitoring system for system-wide power quality measurements. In *Proceedings of the International Conference on Power System Technology (PowerCon 2000)*, volume 2, pages 953–958, 4–7 Dec. 2000.

[33] A. Ciuffoletti. Sharing a common time reference in a heterogeneous distributed system. In *Proceedings of the Seventh Euromicro Workshop on Parallel and Distributed Processing (PDP '99)*, pages 359–366, 1999.

[34] Commodore Amiga, Inc. *Amiga 500/2000 Technical Reference Manual.*

[35] Cypress Semiconductor, 3901 North First St, San Jose, CA 95134, USA. *High Performance Data Book*, 1 Aug. 1993.

[36] W. Dalmau. Critical remarks on the use of medieval eclipse records for the determination of long-term changes in the earth's rotation. *Surveys in Geophysics*, 18(2-3):213–223, 1997. ISSN 0169-3298. Review.

[37] P. Daly, N. Koshelyaevsky, W. Lewandowski, G. Petit, and C. Thomas. Comparison of GLONASS and GPS time transfers. *Metrologia*, 30(2):89–94, May 1993. ISSN 0026-1394.

[38] T. Dawson. *Linux NET-3-HOWTO.* Linux Documentation Project, 29 March 1997. Version 1.1.

[39] M. B. Dewe, S. Sankar, and J. Arrillaga. The application of satellite time references to HVDC fault location. *IEEE Transactions on Power Delivery*, 8(3):1295–1302, 1993. ISSN 0885-8977.

[40] DIN 40050-9*: Straßenfahrzeuge; IP-Schutzarten; Schutz gegen Fremdkörper, Wasser und Berühren; Elektrische Ausrüstung.* Deutsches Institut für Normung e.V., 1993. Ausgabe:1993-05.

[41] Z.-P. Du. Harmonic state estimation of power systems. Master's thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, 1994. xx+140 pp.

[42] B. Fardanesh, S. Zelingher, A. S. Meliopoulos, G. Cokkinides, and J. Ingleson. Multifunctional synchronized measurement network. *IEEE Computer Applications in Power*, 11(1):26–30, 1998. ISSN 0895-0156.

[43] URL `http://www.fftw.org/`. FFTW (Fastest Fourier Transform in the West) is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data.

[44] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Acoustics, Speech, and Signal Processing, ICASSP 1998. Proceedings of the International Conference on*, volume 3, pages 1381–1384, 12–15 May 1998.

[45] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. ISSN 0018-9219.

[46] H. Fruehauf, editor. *Precision Time & Frequency Handbook.* Ball Corporation, Efratom Time & Frequency Products, ninth edition, 1993. 3 Parker, Irvine, CA 92718-1605, USA; (714) 770-5000, fax (714) 770-2463; Fichtenstraße 25, 85649 Hofolding, Germany; 8104/9040, fax 8104/9918.

[47] Gcc, the GNU compiler collection. URL `http://gcc.gnu.org/`.

[48] E. Gershon, N. Mammen, and S. Sidman. *Driving NTSC Monitors Over Fiber-Optic Links Using the* TAXI *chip Set.* Advanced Micro Devices, Inc., 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA, Mar. 1990. 14299A/0-3/90.

[49] B. Guinot. Scales of time. *Metrologia*, 31(6):431–440, 1995. ISSN 0026-1394.

[50] W. A. Halang and M. Wannemacher. High accuracy concurrent event processing in hard real-time systems. *Real-Time Systems*, 12(1):77–94, 1997. ISSN 0922-6443.

[51] R. Hart. *Linux PPP-HOWTO*. Linux Documentation Project, 26 August 1996. Version 2.2.

[52] M. P. Hayes. A real-time beam steering system for KiwiSAS-IV. In *Proceedings of Electronics New Zealand Conference (ENZCON2003)*, pages 27–32. Hamilton, New Zealand, Sept. 2003. URL `http://www.elec.canterbury.ac.nz/research/acoustics/papers/2003/Hayes_enzcon_2003a.pdf`.

[53] ARM and Hellosoft demonstrate low cost VoIP phone on single DSP-enhanced ARM core, 22 Apr. 2003. URL `http://www.hellosoft.com/news/press/press11.htm`. Press release. Also reported by `http://www.arm.com/news/332.html` and `http://www.linuxdevices.com/news/NS9811544680.html`.

[54] P. Hetzel. The low-frequency transmitter DCF77 at 77.5 khz: On 40 years of times signals and standard frequencies, 25 years of coded time information. *PTB-Mitteilungen*, 109(1): 11–18, 1999. ISSN 0030-834X.

[55] P. Hetzel, T. Polewka, A. Söring, and R. Metzger. Time from European time keeping laboratories via telephone modem. In *Tenth European Frequency and Time Forum (EFTF 96)*, pages 540–547, 1996. (IEE Conf. Publ. 418).

[56] K. Horiai. Accuracy of time comparison derived from reception of LORAN-C and global positioning system time signals. *Radio Science*, 27(5):545–551, Sept.-Oct. 1992. ISSN 0048-6604.

[57] S. G. Hunt. The design of a domestic energy management system using a distributed control network. Master's thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, July 1995.

[58] IEC 60529: *Degrees of Protection Provided by Enclosures (IP Code)*. International Electrotechnical Commission, 27 Feb. 2001. URL `http://www.iec.ch/`. Standard, consol. ed. 2.1.

[59] IEC 61000-3-2: *Electromagnetic Compatibility (EMC). Part 3-2: Limits. Limits for Harmonic Current Emissions (Equipment Input Current ≤16 A per Phase)*. International Electrotechnical Commission, 10 Nov. 2004. URL `http://www.iec.ch/`. Standard, consol. ed. 2.2.

[60] IEC/TR 61000-3-7: *Electromagnetic Compatibility (EMC). Part 3: Limits. Section 7: Assessment of Emission Limits for Fluctuating Loads in MV and HV Power Systems—Basic EMC Publication*. International Electrotechnical Commission, 06 Nov. 1996. URL `http://www.iec.ch/`. Standard.

[61] IEC 61000-4-30: *Electromagnetic Compatibility (EMC). Part 4-30: Testing and Measurement Techniques—Power Quality Measurement Methods*. International Electrotechnical Commission, 11 Feb. 2003. URL `http://www.iec.ch/`. Standard, ed. 1.0.

[62] IEC 61000-4-7: *Electromagnetic Compatibility (EMC). Part 4-7: Testing and Measurement Techniques. General Guide on Harmonics and Interharmonics Measurements and Instrumentation, for Power Supply Systems and Equipment Connected Thereto*. International Electrotechnical Commission, 8 Aug. 2002. URL `http://www.iec.ch/`. Standard, ed. 2.0.

[63] IEC/TR 61850-1: *Communication Networks and Systems in Substations. Part 1: Introduction and Overview*. International Electrotechnical Commission, 28 Apr. 2003. URL `http://www.iec.ch/`. Standard.

[64] IEC 61850-5: *Communication Networks and Systems in Substations. Part 5: Communication Requirements for Functions and Device Models*. International Electrotechnical Commission, 24 July 2003. URL `http://www.iec.ch/`. Standard, ed. 1.0.

[65] *IEEE Standard for a Versatile Backplane Bus: VMEbus.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 1987. ISBN 0-471-61601-X. viii+223 pp. ANSI/IEEE Standard 1014-1987.

[66] *IEEE Standard for VMEBus Extensions for Instrumentation: VXIBus.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 1993. ISBN 1-55937-260-5. vi+199 pp. IEEE Standard 1155-1992.

[67] *IEEE Recommended Practice for Monitoring Electric Power Quality.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 1995. ISBN 1-55937-549-3. vi+70 pp. IEEE Standard 1159-1995.

[68] IEEE standard for a high-performance synchronous 32-bit bus: Multibus II. IEEE Standard. IEEE Standards, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, USA.

[69] *IEEE Standard for a High-Performance Synchronous 32-Bit Bus: Multibus II.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 1987. ISBN 0-471-50065-8. 260 pp. LCCN 88-081186. ANSI/IEEE Standard 1296-1987.

[70] *Information Technology—Microprocessor Systems—High-Performance Synchronous 32-Bit Bus: Multibus II.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 1994. ISBN 1-55937-368-7. ii+viii+130 pp. ANSI/IEEE Standard 1296-1994, ISO/IEC 10861:1994.

[71] *IEEE Standard for Synchrophasors for Power Systems.* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 2001. ISBN 1-55937-693-7. vi+30 pp. IEEE Standard 1344-1995(R2001).

[72] *IEEE Recommended Practice for Data Communications Between Remote Terminal Units and Intelligent Electronic Devices in a Substation.* The Institute of Electrical and Electronics Engineers, Inc., 2001. ISBN 0-7381-2639-X. vi+54 pp. IEEE Standard 1379-2000.

[73] *IEEE Standard for a Common Mezzanine Card (CMC) Family.* 2001. IEEE Standard 1386-2001.

[74] *IEEE Standard Physical and Environmental Layers for PCI Mezzanine Cards (PMC).* The Institute of Electrical and Electronics Engineers, Inc., New York, USA, 2001. ISBN 0-7381-2830-9, pdf: 0-7381-2831-7. vi+8 pp. IEEE Standard 1386.1-2001.

[75] *IEEE Standard for a High Performance Serial Bus.* 1996. IEEE Standard 1394-1995.

[76] *IEEE Standard for a High Performance Serial Bus—Amendment 1.* 2000. IEEE Standard 1394a-2000 (Amendment to IEEE Std 1394-1995).

[77] *IEEE Standard for a High Performance Serial Bus—Amendment 2.* 2002. IEEE Standard 1394b-2002 (Amendment to IEEE Std 1394-1995).

[78] *IEEE Standard for Higher Performance Protocol for the Standard Digital Interface for Programmable Instrumentation.* The Institute of Electrical and Electronics Engineers, Inc., 2003. ISBN —. 143 pp. IEEE Standard 488.1-2003.

[79] *IEEE Standard Codes, Formats, Protocols, and Common Commands for Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.* The Institute of Electrical and Electronics Engineers, Inc., 1992. ISBN —. IEEE Standard 488.2-1992.

[80] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95051, USA. *Multibus II Bus Architecture Specification Handbook*, 1984. Order no. 146077-C.

[81] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95051, USA. *Multibus II Interconnect Interface Specification*, 1987, 1988, 1990. Order no. 149299-003.

[82] Intel Corporation, Literature Sales, P.O. Box 58130, Santa Clara, CA 95052-8130, USA. *Simple I/O Design Example using the Multibus II Modular Interface eXtension (MIX) Architecture*, May 1989. Application note AP-433, order no. 281004-001.

[83] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95052-8126, USA. *iSBC CSM/002 Hardware Reference Manual*, 1989, 1990. Order no. 459706-002.

[84] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95051, USA. *Modular Interface eXtension (MIX) Module Devlopment Kit User's Guide*, 1989, 1990. Order no. 500731-003.

[85] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95051, USA. *The MultibusII Peripheral Interface (MPI) User's Manual*, 1990. Order no. 504557-001.

[86] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95052-8126, USA. *Firmware User's Guide for MultibusII System Architecture (MSA) Firmware*, 1990, 1991. Order no. 506090-002.

[87] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95052-8131, USA. *iSBC486/133SE Single Board Computer User's Guide*, 1991. Order no. 506085-001.

[88] Intel Corporation, 3065 Bowers Ave, Santa Clara, CA 95052-8126, USA. *Modular Interface eXtension (MIX) 386/020A and 486/020A Baseboard User's Guide*, 1991. Order no. 468229-001.

[89] Intel Corporation, 2200 Mission College Drive, Santa Clara, CA 95052-8119, USA. *Modular Interface eXtension (MIX) Module Design Specification*, 1991, 1992. Order no. 500729-002.

[90] Intel Corporation, Mt Prospect, IL 60056-7641, USA. *TCP/IP for the iRMX Operating System*, Nov. 1993. Order no. 611476-002.

[91] Intel Corporation, Literature Distribution Center, P.O. Box 7641, Mt Prospect, IL 60056-7641, USA. *Modular Interface eXtension (MIX) 486 Baseboard User's Guide*, Feb. 1994. Order no. 616041-001.

[92] L. V. Kale. Programming languages for CSE: The state of the art. *IEEE Computational Science and Engineering Magazine*, 5(2):18–26, 1998. ISSN 1070-9924. [see also Computing in Science and Engineering].

[93] J. Kato, A. Shimizu, and S. Goto. Active measurement and analysis of delay time in the internet. In *Proceedings of the International Workshops on Parallel Processing*, pages 254–259, 1999.

[94] T. Keppler. *Flicker Measurement and Propagation in Power Systems.* PhD thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, 1998. xxi+181 pp.

[95] T. Keppler, N. Watson, and J. Arrillaga. Computation of the short-term flicker severity index. *IEEE Transactions on Power Delivery*, 15(4):1110–1115, 2000. ISSN 0885-8977.

[96] T. Keppler, N. Watson, J. Arrillaga, and S. Chen. Theoretical assessment of light flicker caused by sub- and interharmonic frequencies. *IEEE Transactions on Power Delivery*, 18(1): 329–333, 2003. ISSN 0885-8977.

[97] T. Keppler, N. R. Watson, S. Chen, and J. Arrillaga. Digital flickermeter realisations in the time and frequency domains. In Proceedings of the Australasian Universities Power Engineering Conference (AUPEC 2001) [180], page ?? ISBN 1-76067-068-X, CD: 1-76067-067-1.

[98] M. Kezunović and B. P. cić. Automated transmission line fault analysis using synchronized sampling at two ends. *IEEE Transactions on Power Systems*, 11(1):441–447, 1996. ISSN 0885-8950.

[99] M. Kihara, A. Imaoka, M. Imae, and K. Imamura. Two-way time transfer through 2.4 gb/s optical SDH system. *IEEE Transactions on Instrumentation and Measurement*, 50(3):709–715, 2001. ISSN 0018-9456.

[100] G. Klawitter. *Zeitzeichensender / Time Signal Stations.* Siebel Verlag, Meckenheim, Germany, 11th edition, 1988. ISBN 3-922221-29-7.

[101] V. Kuhlmann. Fault location on HVdc lines. Master's report, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Feb. 1993. x+86 pp.

[102] V. Kuhlmann. *DSM Hardware Design Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[103] V. Kuhlmann. *DSM Hardware Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00b edition, Nov. 1994.

[104] V. Kuhlmann. *DSM Software Design Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[105] V. Kuhlmann. *DSM Software Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00b edition, Nov. 1994.

[106] V. Kuhlmann. *Synchronous Sampling with CHART.* University of Canterbury, Department of Electrical and Electronic Engineering, Christchurch, New Zealand, 21 June 1996. Application note, version 1.1.

[107] V. Kuhlmann. *Chartdat Manual and Reference.* University of Canterbury, Department of Electrical and Electronic Engineering, Christchurch, New Zealand, June 1997.

[108] V. Kuhlmann, C. P. Arnold, and M. B. Dewe. Relevance of microprocessor advances for power quality instrumentation. Sept. 2005. Submitted, IEEE TPWRD-00582-2005.

[109] V. Kuhlmann, C. P. Arnold, and M. B. Dewe. System requirements of real time continuous data acquisition for power quality monitoring. Sept. 2005. Submitted Elsevier EPSR-D-05-00426.

[110] V. Kuhlmann, M. B. Dewe, and C. P. Arnold. Some aspects of precise synchronisation of data acquisition for power systems harmonic analysis. In Proceedings of the Australasian Universities Power Engineering Conference (AUPEC 2001) [180], pages 84–88. ISBN 1-76067-068-X, CD: 1-76067-067-1. URL `http://volker.top.geek.nz/publications/`.

[111] V. Kuhlmann, M. B. Dewe, and C. P. Arnold. Impact of processor evolution on synchronous measurements. In *Mediterranean Electrotechnical Conference (MELECON 2004)*, pages 1079–1082. Dubrovnik, Croatia, 12–15 May 2004. URL `http://volker.top.geek.nz/publications/`.

[112] V. Kuhlmann, M. B. Dewe, and C. P. Arnold. Implementation considerations in power quality instrumentation design. Sept. 2005. Submitted, IEEE TPWRD-00581-2005.

[113] V. Kuhlmann, A. M. Sinton, M. B. Dewe, and C. P. Arnold. Effects of sampling rate and ADC width on the accuracy of magnitude and phase measurements in power quality monitoring. 2005. Submitted IEEE TPWRD-00173-2005.

[114] J. Kuri. Da geht's lang! Routing, oder: wie die Daten im Internet ihren Weg finden. In *c't Magazin für Computer Technik*, pages 380–389. Verlag Heinz Heise GmbH & Co KG, Hannover, Germany, June 1997. ISSN 0724-8679.

[115] W. S. Kwong, M. J. Clayton, A. Newbould, and J. A. Downes. A microprocessor-based current differential relay for use with digital communication systems: Its design and field experience. In *CIGRE*, Aug. 1986. Paper No. 34.02.

[116] N. P. Laboratory. URL `http://www.npl.co.uk/time/time_scales.html`. GMT.

[117] Lattice Semiconductor Corporation, 5555 Northeast Moore Court, Hillsboro, Oregon 97124, USA. *GAL Data Book*, 1992.

[118] J. Lennon, J. O'Riordan, C. O'Sullivan, P. Twomey, and P. Murphy. Low-cost time synchronisation and tagging using satellite TV teletext broadcasts. *Nuclear Instruments & Methods in Physics Research Section A—Accelerators Spectrometers Detectors and Associated Equipment*, 417(1):169–175, 1998. ISSN 0168-9002.

[119] N. Lethaby and D. Friedland. How to write DSP device drivers. *Embedded Systems Programming*, pages 21–28, Jan. 2004.

[120] J. Levine. Time synchronization using the internet. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 45(2):450–460, 1998. ISSN 0885-3010.

[121] J. Levine. Time synchronization over the internet using an adaptive frequency-locked loop. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 46(4):888–896, 1999. ISSN 0885-3010.

[122] J. Levine. Time and frequency distribution using satellites. *Reports on Progress in Physics*, 65(8):1119–1164, 2002. ISSN 0034-4885. Review.

[123] W. Lewandowski and J. Azoubib. Time transfer and TAI. In *Proceedings of the IEEE/EIA International Frequency Control Symposium and Exhibition*, pages 586–597, 2000.

[124] W. Lewandowski, J. Azoubib, and W. Klepczynski. GPS: Primary tool for time transfer. *Proceedings of the IEEE*, 87(1):163–172, 1999. ISSN 0018-9219.

[125] W. Lewandowski, J. Danaher, and W. Klepczynski. GLONASS common-view time transfer between North America and Europe and its comparison with GPS. In *Tenth European Frequency and Time Forum (EFTF 96)*, pages 388–392, 1996. (IEE Conf. Publ. 418).

[126] W. Lewandowski, G. Petit, and C. Thomas. Precision and accuracy of GPS time transfer. *IEEE Transactions on Instrumentation and Measurement*, 42(2):474–479, 1993. ISSN 0018-9456.

[127] B. Lewis. *CADU Software Design Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[128] B. Lewis. *CADU Software Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[129] H. Li, E. Southern, P. Crossley, S. Potts, S. Pickering, B. Caunce, and G. Weller. A new type of differential feeder protection relay using the global positioning system for data synchronization. *IEEE Transactions on Power Delivery*, 12(3):1090–1099, 1997. ISSN 0885-8977.

[130] Y. Li and C. Zhang. Secular variation of earth's rotation: Inferred from the chinese ancient shoushi calendar (ad 1281). *Earth, Moon, and Planets: an International Journal of Comparative Planetology*, 76(1-2):11–17, 1997-1998. ISSN 0167-9295.

[131] R. Lichtenecker. Terrestrial time signal dissemination. *Real-Time Systems*, 12(1):41–61, 1997. ISSN 0922-6443.

[132] Magnavox Electronic Systems Company, 2829 Maricopa Street, Torrance, California 90503, USA. *MX4200 Installation and Service Manual*, Oct. 1991. R-7122B, No. 60232.

[133] Magnavox Electronic Systems Company, 2829 Maricopa Street, Torrance, California 90503, USA. *MX4200 PC Controller User's Guide*, Oct. 1991. R-7144A, No. 60226.

[134] Magnavox Electronic Systems Company, 2829 Maricopa Street, Torrance, California 90503, USA. *MX4200 GPS Receiver Programmer's Guide*, Feb. 1992. R-7145A, No. 60222.

[135] Magnavox field service bulletin 4200-016: MX4200 software release version B035 & version D035. 2829 Maricopa Street, Torrance, CA 90503, USA, Apr. 1993. 5 pp. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[136] Magnavox field service bulletin 4200-017: MX 4200 software release version B035 & version D035 input & output record field descriptions. 2829 Maricopa Street, Torrance, CA 90503, USA, Apr. 1993. 24 pp. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[137] Magnavox field service bulletin MX4200-001: MX 4200 solid yellow LED. 2829 Maricopa Street, Torrance, CA 90503, USA, Dec. 1990. 2 pp. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[138] Magnavox field service bulletin MX4200-002: MX 4200 reset commands. 2829 Maricopa Street, Torrance, CA 90503, USA, Dec. 1990. 2 pp. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[139] Magnavox field service bulletin MX4200-006: MX 4200 common questions and answers. 2829 Maricopa Street, Torrance, CA 90503, USA, Jan. 1991. 4 pp. Marine and Survey Systems Division; (213) 618-1200; FAX (213) 618-7001.

[140] Magnavox field service bulletin MX4200-013: Guide for interfacing with the MX 4200. 2829 Maricopa Street, Torrance, CA 90503, USA, Feb. 1992. 12 pp. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[141] Magnavox field service bulletin MX4200-015: MX 4200 software version B031 and version D031. 2829 Maricopa Street, Torrance, CA 90503, USA, Apr. 1992. 19 pp. Marine and Survey Systems Division; (310) 618-1200; FAX (310) 618-7001.

[142] Magnavox product support bulletin MX4200-36: MX 4200 software version A35. 23860 Hawthorne Blvd., Torrance, CA 90505, USA, Aug. 1994. 2 pp. (310) 791-2141; FAX (310) 791-6108.

[143] Magnavox product support bulletin MX4200-37: MX 4200 software version B35. 23860 Hawthorne Blvd., Torrance, CA 90505, USA, Aug. 1994. 2 pp. (310) 791-2141; FAX (310) 791-6108.

[144] F. Meyer. One-way time transfer using geostationary satellite TDF2. *IEEE Transactions on Instrumentation and Measurement*, 44(2):103–106, 1994. ISSN 0018-9456.

[145] G. Meyer and K. Manstorfer. Non-central generation of audio-frequency signals for audio-frequency ripple control systems. *Elektrizitätswirtschaft*, 92(10):630–631, 1993. ISSN 0013-5496.

[146] Micro Channel Developers Association. *Micro Channel Architecture Specification*, Jan. 1992. Part-no. MCDA0001-192, revision 1.0.

[147] A. Miller, S. Chen, D. Chuah, and U. Zuckerman. CHART *System Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[148] A. J. V. Miller. *Power System Harmonic Monitoring.* PhD thesis, Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, Dec. 1992.

[149] A. J. V. Miller. *DAPM Hardware Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[150] A. J. V. Miller. *DAPM Hardware Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, 1994. Not released.

[151] A. J. V. Miller. *DAPM Software Design Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, 1994. Not released.

[152] A. J. V. Miller. *DAPM Software Functional Requirement Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994. Not released.

[153] A. J. V. Miller, S. Chen, and U. Zuckerman. *MIX Baseboard Software Design Specification.* University of Canterbury, Christchurch, New Zealand, 1.00a edition, Nov. 1994.

[154] A. J. V. Miller, D. Chuah, M. B. Dewe, and V. Kuhlmann. CHART *System Design Overview.* University of Canterbury, Christchurch, New Zealand, 1.00b edition, Nov. 1994.

[155] A. J. V. Miller and M. B. Dewe. The application of multirate digital signal processing techniques to the measurement of power system harmonic levels. In *Proceedings of the 5th International Conference on Harmonics in Power Systems.* IEEE Power Engineering Society, Sept. 1992. Atlanta / Georgia, USA; 23–25 Sep 92.

[156] A. J. V. Miller and M. B. Dewe. Multichannel continuous harmonic analysis in real-time. *IEEE Transactions on Power Delivery*, 7(4):1813–1819, Oct. 1992.

[157] D. Mills. Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI. RFC 2030, Oct. 1996. URL `ftp://ftp.ietf.org/rfc/rfc2030.txt`.

[158] D. L. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991. ISSN 0090-6778.

[159] D. L. Mills. Network time protocol (version 3) specification, implementation. RFC 1305, Mar. 1992. URL `ftp://ftp.ietf.org/rfc/rfc1305.pdf`.

[160] D. L. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Transactions on Networking*, 3(3):245–254, 1995. ISSN 1063-6692.

[161] D. L. Mills. Adaptive hybrid clock discipline algorithm for the network time protocol. *IEEE/ACM Transactions on Networking*, 6(5):505–514, 1998. ISSN 1063-6692.

[162] G. W. Mobbs and J. D. Lynch. TAXI *Error Management.* Advanced Micro Devices, Inc., 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA, Jan. 1990. 14229A/0-1/90.

[163] Motorola, 8201 E. McDowell Road, Scottsdale, AZ 85252, USA. *SixGun Model 610/620 Series DGPS Receivers PC Controller Software Guide*, Aug. 1993. 68P38368E Rev. 4.2.

[164] Motorola, 8201 E. McDowell Road, Scottsdale, AZ 85252, USA. *SixGun Model 610/620 Series DGPS Receivers Operator's Manual*, Apr. 1994. 68-P38358E Rev. 3.6.

[165] Motorola Inc., Phoenix, Arizona 85036, USA. *MC68HC16Z1 User's Manual.* MC68HC16Z1UM/AD.

[166] Motorola Inc., Phoenix, Arizona 85036, USA. *MC68HC16Z1 16-Bit Modular Microcontroller Technical Summary*, 1992. MC68HC16Z1TS/D Rev. 2, replaces BR754/D Rev. 1.

[167] Motorola Inc., Phoenix, Arizona 85036, USA. *CPU16 Central Processor Unit Reference Manual*, 1993. CPU16RM/AD Rev 2.

[168] S. Mueller. *Upgrading and Repairing PCs.* Que, 16th edition, Sept. 2005. ISBN 0-7897-3173-8.

[169] R. Nelson, D. McCarthy, S. Malys, J. Levine, B. Guinot, H. Fliegel, R. Beard, and T. Bartholomew. The leap second: Its history and possible future. *Metrologia*, 38(6):509–529, 2001. ISSN 0026-1394. Review.

[170] J. Neter, W. Wasserman, and G. A. Whitmore. *Applied Statistics.* Allyn and Bacon, Newton, Massachusetts, 3rd edition, 1988. ISBN 0-205-10328-6.

[171] *Standard for Interfacing Marine Electronics.* National Marine Electronics Association. URL `http://www.nmea.org/pub/0183/`. Standard NMEA 0183.

[172] J. Odagiri, N. Yamamoto, and T. Katoh. Porting EPICS to L4-Linux based system. In *Proceedings of the 2001 Particle Accelerator Conference (PAC 2001)*, volume 2, pages 1183–1185, 2001.

[173] Y. Ohura, M. Suzuki, K. Yanagihashi, M. Yamaura, K. Omata, T. Nakamura, S. Mitamura, and H. Watanabe. A predictive out-of-step protection system based on observation of the phase difference between substations. *IEEE Transactions on Power Delivery*, 5(4):1695–1704, 1990. ISSN 0885-8977.

[174] A. G. Phadke. Synchronized phasor measurements in power systems. *IEEE Computer Applications in Power*, 6(2):10–15, 1993. ISSN 0895-0156.

[175] A. G. Phadke, B. Pickett, M. Adamiak, M. Begovic, G. Benmouyal, J. R. O. Burnett, T. W. Cease, J. Goossens, D. J. Hansen, M. Kezunovic, L. L. Mankoff, P. G. McLaren, G. Michel, R. J. Murphy, J. Nordstrom, M. S. Sachdev, H. S. Smith, J. S. Thorp, M. Trotignon, T. C. Wang, and M. A. Xavier. Synchronized sampling and phasor measurements for relaying and control. *IEEE Transactions on Power Delivery*, 9(1):442–452, Jan. 1994. ISSN 0885-8977.

[176] Philips Semiconductors. *74ABT245 Octal Transceiver with Direction Pin (3-State)*, 21 June 1993. 653-1447 10073.

[177] Philips Semiconductors. *IC's for Data Communication*, May 1994.

[178] L. Prechelt. An empirical comparison of seven programming languages. *IEEE Computer*, 33 (10):23–29, 2000. ISSN 0018-9162.

[179] *Proceedings of the 7th International Conference on Harmonics and Quality of Power (ICHQP).* Las Vegas, NV, USA, 16–18 Oct. 1996. ISBN none.

[180] *Proceedings of the Australasian Universities Power Engineering Conference (AUPEC 2001).* Curtin University of Technology, Perth, Australia, 23–26 Sept. 2001. ISBN 1-76067-068-X, CD: 1-76067-067-1.

[181] C. L. Pryke. Long baseline testing of a GPS based, autonomous event time-tagging system. *Nuclear Instruments & Methods in Physics Research Section A—Accelerators Spectrometers Detectors and Associated Equipment*, 394(3):420–422, 1997. ISSN 0168-9002. Letter.

[182] C. L. Pryke and J. Lloyd-Evans. A high-performance GPS based autonomous event time-tagging system with application in a next-generation extensive-air-shower array. *Nuclear Instruments & Methods in Physics Research Section A—Accelerators Spectrometers Detectors and Associated Equipment*, 354(2-3):560–566, 1995. ISSN 0168-9002.

[183] PTB, editor. *Zur Zeit.* Physikalisch Technische Bundesanstalt (PTB), Braunschweig, Germany, März 1988.

[184] Quality Technologies. *74OL6010 High Speed Opto-Coupler With Logic Interface.* Short data.

[185] J. R. O. Burnett, M. M. Butts, T. W. Cease, V. Centeno, G. Michel, R. J. Murphy, and A. G. Phadke. Synchronized phasor measurements of a power system event. *IEEE Transactions on Power Systems*, 9(3):1643–1650, Aug. 1994. ISSN 0885-8950.

[186] J. R. O. Burnett, M. M. Butts, and P. S. Sterlina. Power system applications for phasor measurement units. *IEEE Computer Applications in Power*, 7(1):8–13, 1994. ISSN 0895-0156.

[187] J. Radke and R. Finger. *The Multibus II Interconnect Design Guide.* Intel Corporation, Literature Sales, P.O. Box 58130, Santa Clara, CA 95052-8130, USA, 1 Jan. 1988. Application note AP-423, order no. 280640-001.

[188] A. Radovanovic. Using the internet in networking of synchronized phasor measurement units. *International Journal of Electrical Power & Energy Systems*, 23(3):245–250, 2001. ISSN 0142-0615.

[189] Rakon Ltd, Auckland, New Zealand. *VTXO4080 Voltage Controlled Crystal Oscillator.* Data sheet.

[190] P. E. Ross. Commandments. The rules engineers live by weren't always set in stone. *IEEE Spectrum*, 40(12):30–35, Dec. 2003. ISSN 0018-9235.

[191] A. Sachenko, V. Kochan, R. Kochan, V. Turchenko, K. Tsahouridis, and T. Laopoulos. Error compensation in an intelligent sensing instrumentation system. In *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference (IMTC 2001)*, volume 2, pages 869–874, 2001.

[192] A. Sachenko, V. Kochan, and V. Turchenko. Instrumentation for gathering data. designing intelligent distributed data acquisition systems that are accurate, reliable, and adaptable. *IEEE Instrumentation and Measurement Magazine*, 6(3):34–40, Sept. 2003. ISSN 1094-6969.

[193] S. Sankar. DC line fault locator — EMTP studies. Technical report, University of Canterbury, Department of Electrical and Electronic Engineering, Christchurch, New Zealand, Nov. 1991.

[194] U. Schmid and K. Schossmaier. Interval-based clock synchronization. *Real-Time Systems*, 12 (2):173–228, 1997. ISSN 0922-6443.

[195] K. Schossmaier, U. Schmid, M. Horauer, and D. Loy. Specification and implementation of the universal time coordinated synchronization unit (UTCSU). *Real-Time Systems*, 12(3): 295–327, May 1997. ISSN 0922-6443.

[196] J. Schoukens, F. Louage, and Y. Rolain. Study of the influence of clock instabilities in synchronized data acquisition systems. *IEEE Transactions on Instrumentation and Measurement*, 45 (2):601–604, 1996. ISSN 0018-9456.

[197] P. Scott. TAXI *chip 4B/5B Codes.* Advanced Micro Devices, Inc., 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA, Apr. 1990. 14170A/0-4/90.

[198] Y. Serizawa, H. Imamura, and M. Kiuchi. Performance evaluation of IP-based relay communications for wide-area protection employing external time synchronization. In *IEEE Power Engineering Society Summer Meeting*, volume 2, pages 909–914, 2001.

[199] Siemens AG. *50 MBaud Transmitter and Receiver (ECL) Modules.* Product information.

[200] Siemens AG. *50 MBaud Transmitter and Receiver (TTL) Modules.* Product information.

[201] Siemens Aktiengesellschaft. *OME — On-Board Module Expansion.*

[202] T. Skeie, S. Johannessen, and Øyvind Holmeide. Highly accurate time synchronization over switched ethernet. In *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, volume 1, pages 195–204, 2001.

[203] J. Sousa, A. Batista, A. Combo, C. Correia, C. A. F. Varandas, D. Trotman, and J. Waterhouse. A distributed system for fast timing and event management on the MAST experiment. *Fusion Engineering and Design*, 43(3-4):407–412, Jan. 1999. ISSN 0920-3796.

[204] J. Sousa, A. Combo, A. Batista, C. Correia, C. Varandas, D. Trotman, and J. Waterhouse. The 32 bit timing unit of a real-time event-based control system for a nuclear fusion experiment. *IEEE Transactions on Nuclear Science*, 45(4):2052–2056, 1998. ISSN 0018-9499.

[205] J. Sousa, A. Combo, A. Batista, M. Correia, D. Trotman, J. Waterhouse, and C. A. F. Varandas. A distributed real-time system for event-driven control and dynamic data acquisition on a fusion plasma experiment. *Fusion Engineering and Design*, 48(1-2):31–36, Aug. 2000. ISSN 0920-3796.

[206] E. P. Southern, H. Y. Li, P. A. Crossley, S. Potts, and B. R. J. Caunce. Development of a GPS synchronized phasor measurement unit using delta-sigma ADCs and DSP. In *Proceedings of the 1996 31st Universities Power Engineering Conference. Part 3 (of 3)*, volume 3, pages 850–853. Technological Educational Institute, Iraklio, Greece, Iraklio, Greece, 18-20 Sept. 1996.

[207] SSH is the secure shell protocol created by Tatu Ylönen, `http://ssh.com`. It provides encrypted login sessions, and has the ability to tunnel any number of TCP/IP connections transparently through the encrypted channel. The freely available implementation from `http://openssh.org/` is in widespread use.

[208] F. Stephenson and L. Morrison. Long-term fluctuations in the earth's rotation—700-BC to AD-1990. *Philosophical Transactions of the Royal Society of London Series A—Mathematical Physical and Engineering Sciences*, 351(1695):165–202, 1995. ISSN 0962-8428.

[209] W. Stuber. The calibration of oscillators using the DCF77 carrier frequency. *Elektronik*, 43 (8):80–92, 19 April 1994. ISSN 0013-5658.

[210] Sun Microsystems, Inc., 2550 Garcia Ave, Mountain View, CA 94043, USA. *The SBus Rule Book*. 5/91 FE314-0/3K.

[211] Texas Instruments Inc., P.O. Box 655303, Dallas, TX 75265, USA. *TIL191, TIL192, TIL193 ... Optocouplers*, Sept. 1989.

[212] Texas Instruments Inc., P.O. Box 655303, Dallas, TX 75265, USA. *SN55LBC176, SN65LBC176, SN75LBC176, Differential Bus Transceivers*, Feb. 1993.

[213] Texas Instruments Incorporated, Houston, Texas, USA. *TMS320C3x C Source Debugger User's Guide*, Sept. 1991. SPRU053A.

[214] Texas Instruments Incorporated, Houston, Texas, USA. *TMS320C3x User's Guide*, 1991. SPRU031B.

[215] Texas Instruments Incorporated, Houston, Texas, USA. *TMS320C31 Embedded Control Technical Brief*, 1992. SPRU083.

[216] Texas Instruments Incorporated, Houston, Texas, USA. *TMS320 Floating-Point DSP Assembly Language Tools User's Guide*, Feb. 1995. SPRU035B.

[217] Texas Instruments Incorporated, Houston, Texas, USA. *TMS320 Floating-Point DSP Optimizing C Compiler User's Guide*, Feb. 1995. SPRU034F.

[218] H. C. Torng and M. Day. Interrupt handling for out-of-order execution processors. *IEEE Transactions on Computers*, 42(1):122–127, 1993. ISSN 0018-9340.

[219] Trimble Navigation Limited, 585 North Mary Avenue, Sunnyvale, CA 94086, USA. *SVeeSix 6-Channel GPS Sensor Specification and User's Manual*, Nov. 1992. 21500-00 Rev. D.

[220] C. Tøndering. Frequently asked questions about calendars, 24 June 2003. URL `http://www.tondering.dk/claus/calendar.html`. Version 2.6.

[221] S. Underwood et al. Porting linux to DSPs and C support, July 1999. URL `http://www.ussg.iu.edu/hypermail/linux/kernel/9907.3/0447.html`. Archived internet discussion.

[222] Universal serial bus. URL `http://www.usb.org/`.

[223] University of Canterbury, Christchurch, New Zealand. *RDCM Hardware Design Specification*, 1.00a edition, 1994. Not released.

[224] M. van As and H. Vermeulen. A GPS based time-stamping and scheduling system for wide area power system measurements. In *IEEE 6th Africon Conference in Africa (AFRICON)*, volume 2, pages 853–857, 2002.

[225] G. D. van Rossum. *Die Geschichte der Stunde: Uhren und moderne Zeitordnungen*. Deutscher Taschenbuch Verlag, München, Dec. 1995. ISBN 3-423-04673-2. The History of the Hour: Clocks and Modern Timekeeping.

[226] J. Vanderkooy and S. P. Lipshitz. Resolution below the least significant bit in digital systems with dither. *Journal of the Audio Engineering Society*, 32(3):106–113, Mar. 1984. Correction in 32(11) p. 889 (Nov 1984).

[227] J. Watkinson. *The Art of Digital Audio*. Focal Press, London, 3rd edition, Dec. 2000. ISBN 0-240-51587-0.

[228] S. Wolf. Preiswerte DCF77 Zeitnormale für verteilte Anwendungen. Exact but inexpensive DCF77 time receivers for distributed applications. *Frequenz*, 51(5/6):148–153, May-June 1997. ISSN 0016-1136.

[229] XILINX, 2100 Logic Drive, San Jose, CA 95124, USA. *Synchronous Presettable Counter*. Application note XAPP 003.002.

[230] XILINX, 2100 Logic Drive, San Jose, CA 95124-3400, USA. *The Programmable Logic Data Book*, 2 edition, 1994. PN0401253.

[231] XILINX, 2100 Logic Drive, San Jose, CA 95124-3400, USA. *XACT User Guide*, Apr. 1994.

[232] H.-S. Yan and T.-Y. Lin. A study on ancient chinese time laws and the time-telling system of Su Song's clock tower. *Mechanism and Machine Theory*, 37(1):15–33, Jan. 2002.

[233] C. Yunping, Z. Chengxue, and H. Zhijian. The hardware configuration and field experiment of GPS-based for online parameter tester for transmission lines with mutual inductance. In *Proceedings of the International Conference on Energy Management and Power Delivery (EMPD '98)*, volume 2, pages 381–385, 1998.